Data-Driven Discovery of Mechanistic Ecosystem Models with LLMs

Scott Spillias[1,2*] Jacob Rogers[3] Fabio Boschetti[2,4]
Beth Fulton[1,2] Magda Guglielmo[5] SukYee Yong[5]
Rowan Trebilco[1,2]

July 14, 2025

[1]CSIRO Environment, Hobart, Australia
[2]Centre for Marine Socio-Ecology, University of Tasmania, Hobart, Australia
[3]CSIRO Environment, St. Lucia, Australia
[4]CSIRO Environment, IOMRC Crawley, Australia
[5]CSIRO 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

*Corresponding author: scott.spillias@csiro.au

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 critically review AIME's ability to recover known ecological relationships through two complementary marine case studies: (1) a nutrient-phytoplankton-zooplankton model, and (2) a Crown-of-Thorns starfish (COTS) model. 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, best AIME generated models approached human expert models in terms of their ability to successfully capture COTS outbreak dynamics and demonstrated strong out-of-sample predictive power. AIME produces interpretable models with meaningful parameters that capture real biological processes, facilitating scientific insight and potentially accelerating management applications. By dramatically accelerating model development while offering ecological interpretability, AIME offers a powerful new tool for addressing urgent ecological challenges in rapidly changing environments.

Main

Ecosystem models provide invaluable information for managing complex interactions between nature and people[1,2], but their development traditionally requires significant time and expertise, creating a bottleneck in addressing urgent environmental challenges[3,4], particularly as climate change demands rapid, adaptable approaches for ecosystem management[5,6].

Artificial Intelligence (AI) offers great promise as a part of the solution to these modelling challenges, with potential to accelerate model development and enhance adaptability[7]. While initial efforts to apply AI in ecological modelling focused on machine learning approaches that rely on black-box methods[8], emerging techniques in equation discovery and automated scientific discovery show particular promise[9,10]. These methods can derive interpretable mathematical relationships directly from data, offering advantages over statistical emulators when modelling novel environmental conditions[11–13]. Similarly, attempts to leverage large language models (LLMs) for direct time-series prediction[14–21], 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[20] achieves impressive accuracy in forecasting physical variables like streamflow and water temperature, but does not address the mechanistic relationships needed for ecosystem management and applications.

Rather than using AI to replace traditional modelling approaches, recent advances in AI coding capabilities suggest a more promising direction[22]. Language models like o3-mini and Claude can assist in constructing mechanistic models[7], maintaining interpretability while accelerating the development process[23]. Recent demonstrations of LLMs automating scientific processes, from autonomous chemical experimentation[24] to biomedical research[25], evidence synthesis[26] and even fully automated scientific discovery[27], 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[7,27]. 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 (Figure 1). 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 and offers a way to explore relative support for multiple possible mechanistic structures.

We test AIME's ability to construct plausible ecological models 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 understanding using synthetic data generated from a well-established nutrient-phytoplankton-zooplankton (NPZ) model[28]. 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 understanding. Second, we assess AIME's ability to provide management-relevant predictions using synthetic data based on Crown-of-Thorns starfish (COTS) populations on the Great Barrier Reef, derived from existing MICE models[29–32]. 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 COTS immigrants), tests the framework's robustness while focusing on a specific management challenge: quantifying outbreaks in a complex predator-prey system. Through systematic comparison of different LLMs (o3-mini, o4-mini, and GPT4.1 from OpenAI, Claude Sonnet 3.7 and 3.7 from Anthropic, and Gemini 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 are potentially suitable for use in support of ecosystem management.

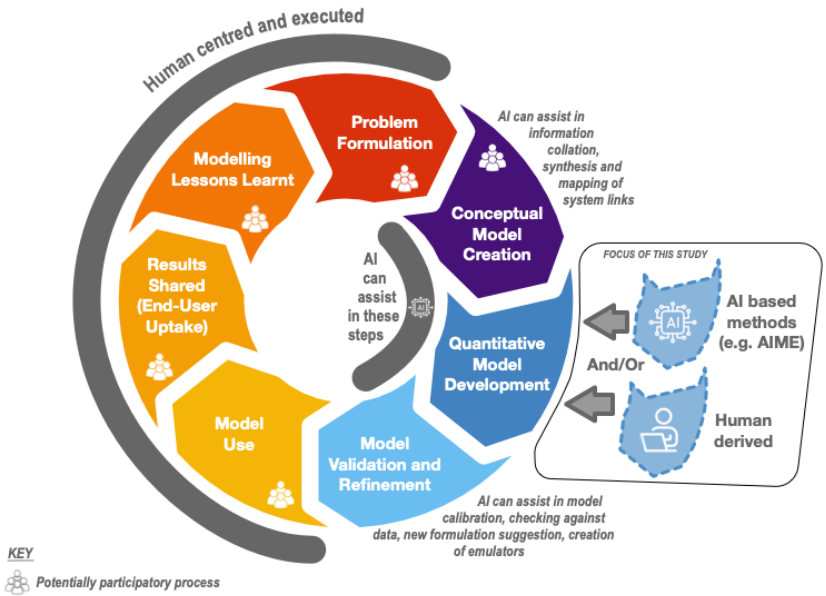

Figure 1: Stylised representation of the iterative modelling process that AIME aims to support. Whilst human experts drive the majority of the process, we show that AI-driven processes could play an important role in the Model Development stage.

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.

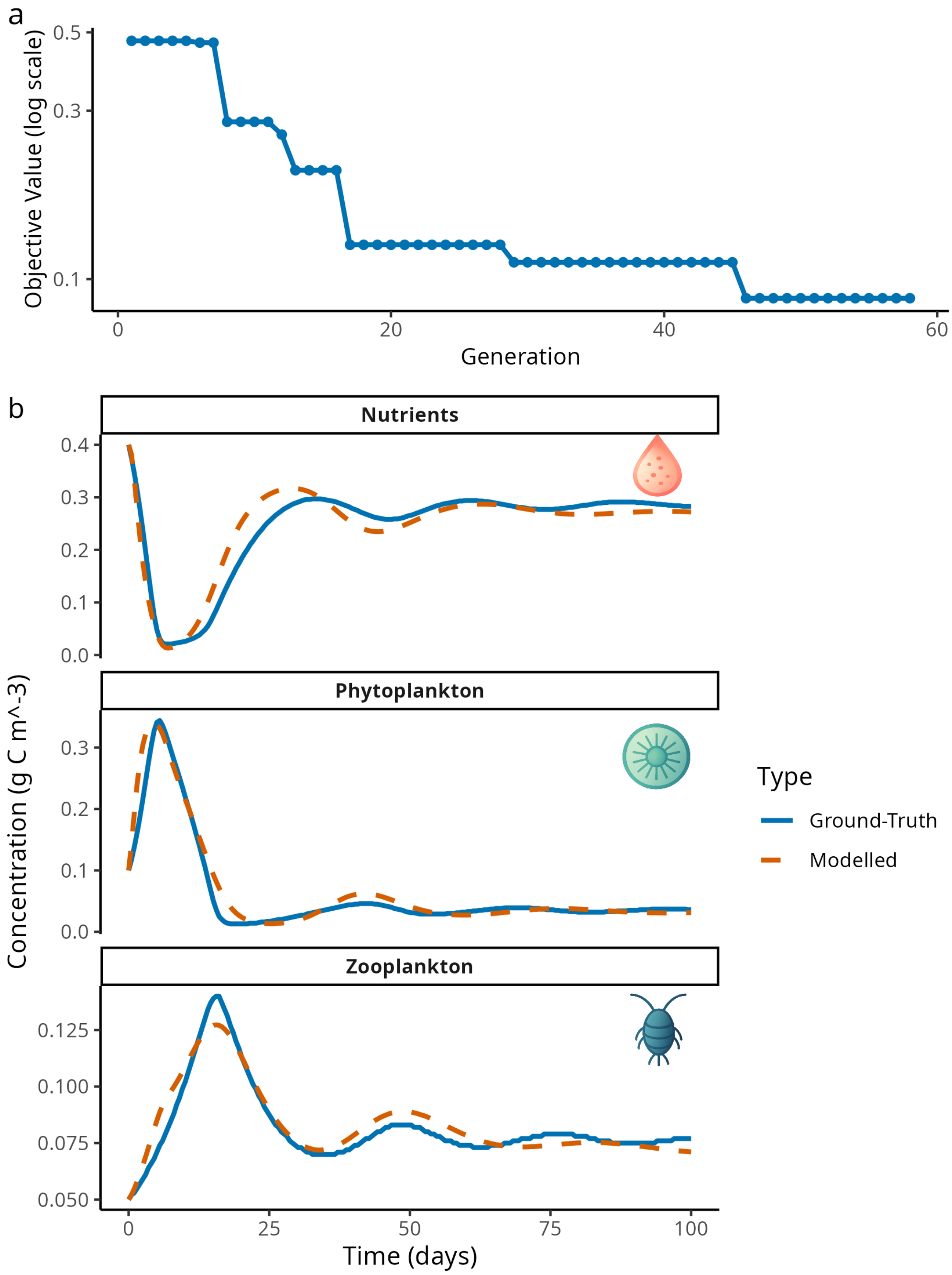

Figure 2: Evolution and performance of the best NPZ model. (A) Training progress showing the objective value across generations on a log-scale. (B) Time series comparison between ground-truth and modelled NPZ dynamics for the best-performing model (objective value = 0.0883). The plots show the temporal 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 8 in Supplementary Materials) revealed that some mechanisms were more readily recovered than others.

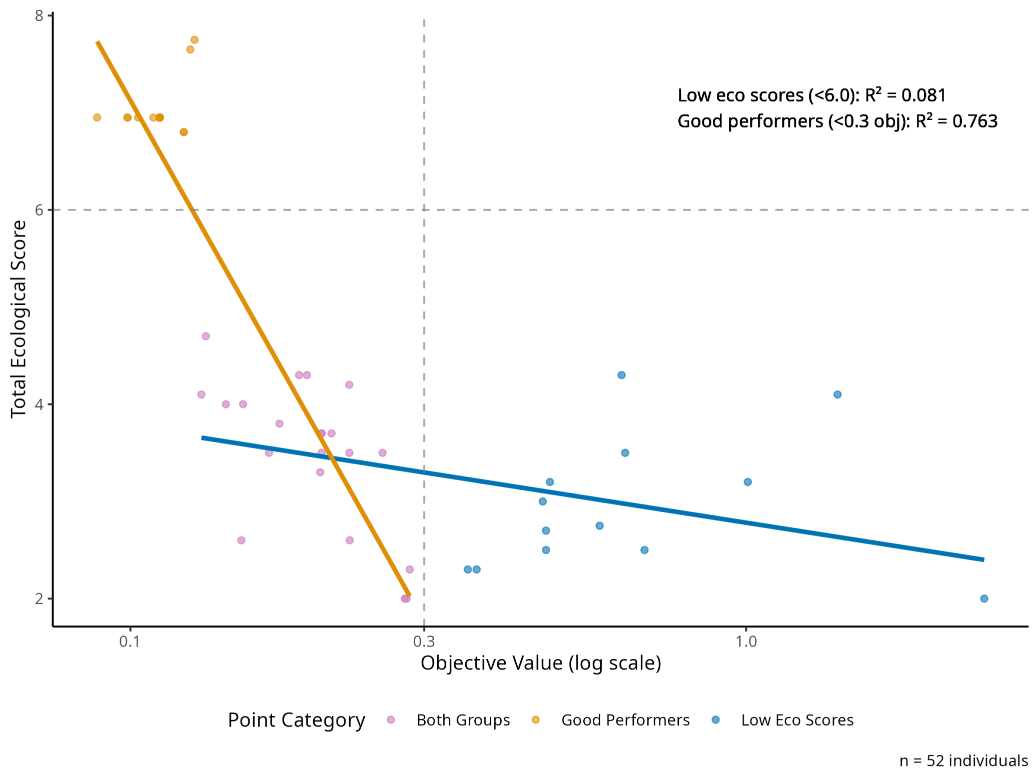

Figure 3: Log-log Relationship between total ecological accuracy score and model performance (objective value). Lower objective values indicate better model fit, while higher ecological scores indicate closer alignment with known NPZ model mechanisms. The data distribution appears to cluster into two 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.

COTS Case Study

For all but one LLM, AIME was able to generate ecosystem models with prediction accuracy that was approaching the quantitative fit of the expert-developed model (objective value (NMSE): 0.2312). After ten generations, objective values across LLMs were as follows: gpt4.1 achieved the best objective value (0.3488; which was achieved in the first generation and unimproved over subsequent generations), followed by Claude Sonnet-3.7 (0.5204), then o3-mini (0.5606), o4-mini (0.5786), and finally Claude Sonnet-3.6 (0.6599).

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).

The ecological models produced by AIME exhibited substantial structural diversity in their ecological mechanisms and usability, while sharing several foundational elements. Across all models, density-dependent COTS population growth, resource limitation based on coral availability, and differential impacts on fast versus slow-growing coral species emerged as common features.

Temperature effects were implemented through markedly different approaches across the LLMs. Claude 3.6 and Claude 3.7 Sonnet employed Gaussian response curves where COTS survival peaks at optimal temperatures. In contrast, the o3-mini and gpt 4.1 models utilized linear temperature factors that directly modify growth rates. The o4-mini model took yet another approach, incorporating a number of abstract and vague parameters such as 'environemtal modifier' and explicit outbreak parameters without explicitly modelling temperature.

The representation of COTS-coral interactions similarly varied across models. The o3 mini LLM implemented resource limitation with quadratic adjustment factors for outbreak triggering, creating non-linear responses to coral availability. Claude 3.6 Sonnet featured temperature-dependent recruitment with resource limitation based on total coral cover. Claude 3.7 Sonnet employed Holling Type II functional responses for predation with explicit food limitation effects that increase COTS mortality when coral cover is low. The o4 mini model utilized resource limitation with saturating functions and quadratic terms to capture diminishing returns in resource uptake. Finally, the gpt 4.1 model incorporated differential predation on coral types with varying assimilation efficiencies.

There were notable differences in the behaviour and speed of convergence for all of the LLMs we tested Figure S5. The o3-mini model completed ten generations in as little as 41 minutes. The two Claude-Sonnet LLMs and OpenAI's GPT4.1 were able to generate well-functioning models in a single generation, but struggled to improve upon previous generations, whereas o3-mini and o4-mini (the two 'reasoning' models we included in this study) were able to consistently improve upon previous generations. Due to this behaviour, we decided to run o4-mini for an additional 90 generations (100 generations total), to see if it would converge at or near the human model.

Ultimately, it reached an objective value of 0.3486, but was unable to fully capture the outbreak dynamics of the CoTS as requested in the research topic (Figure 4). We note that, for the purposes of this initial exploration, the objective function was weighted equally for all time-series, whereas the human modeller may have prioritised capturing the two outbreaks exhibited in the COTS time-series.

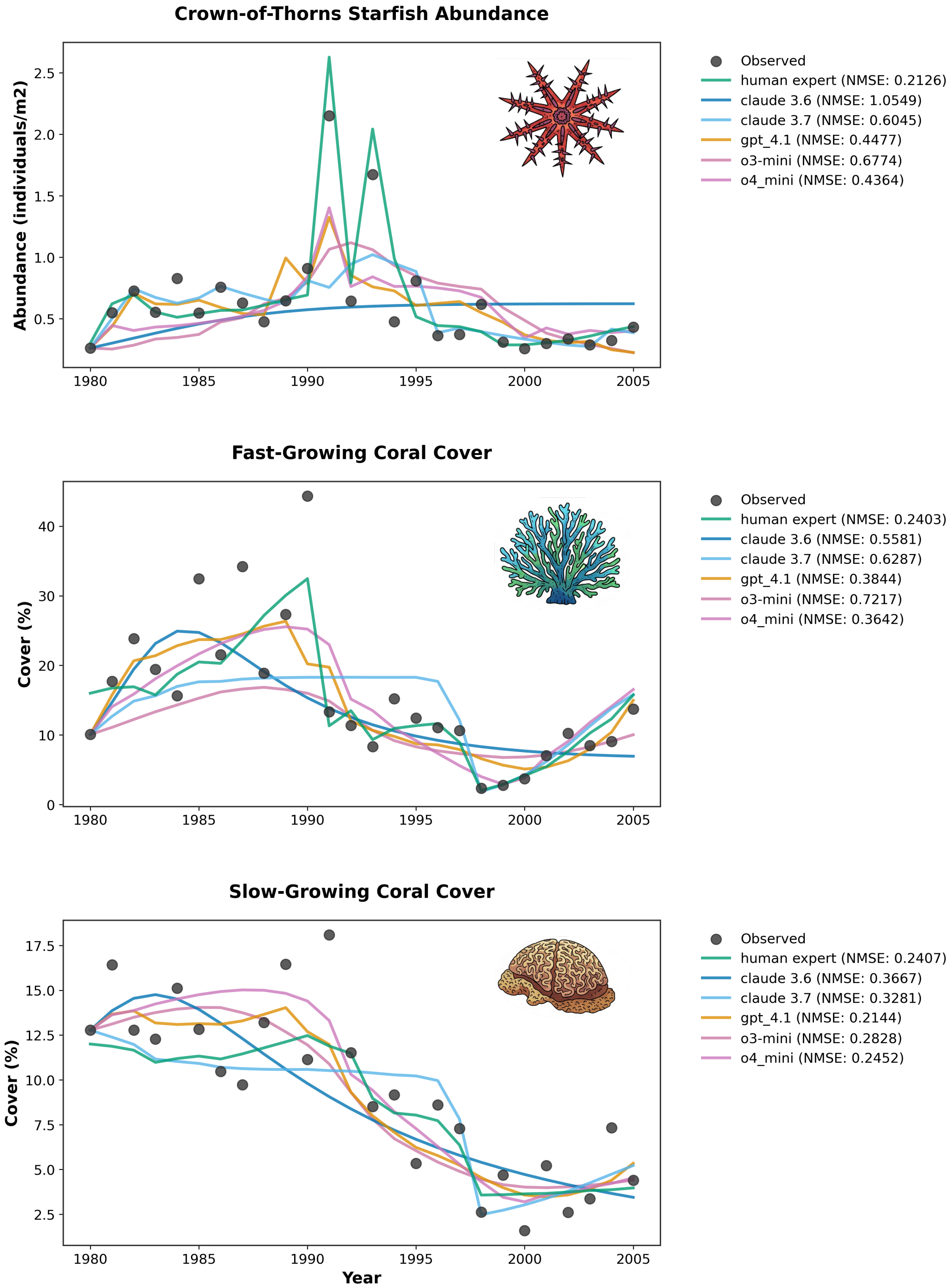

Figure 4: Comparison of model predictions across ecosystem components. The plots show observed versus predicted values for COTS abundance, fast-growing coral cover, and slow-growing coral cover, demonstrating the models' ability to capture key ecological patterns and relationships. Objective values (obj) 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

Because the o4-mini LLM demonstrated a high capability to improve quantitative performance across generations, we used it as the base LLM to test whether the AIME framework could create models that are capable of predicting out of sample datapoints. After ten generations, we found that the best performing model (objective value: 0.4677) had a reasonable degree of out-of-sample predictions performance on the withheld test dataset, with particularly strong predictive power for fast-growing coral cover ($R^2 = 0.768$, RMSE = 2.169, MAE = 2.025).

For slow-growing coral cover, the model achieved moderate predictive accuracy ($R^2 = 0.187$, RMSE = 1.929, MAE = 1.568), effectively capturing the general declining trend while showing some deviation in precise values. COTS population predictions demonstrated reasonable accuracy (R^2 value 0.569, RMSE = 0.073, MAE = 0.064) despite not fully capturing the magnitude of peaks in COTS abundance in the training data.

Figure 5 illustrates these prediction capabilities, showing both training period performance (pre-1997) and out-of-sample predictions (1997-2005). Interestingly, in contrast to the o4-mini's fairly vague equations described above, this iteration featured more tangible parameter and equations formulations (See model details in Section S8). 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 this system.

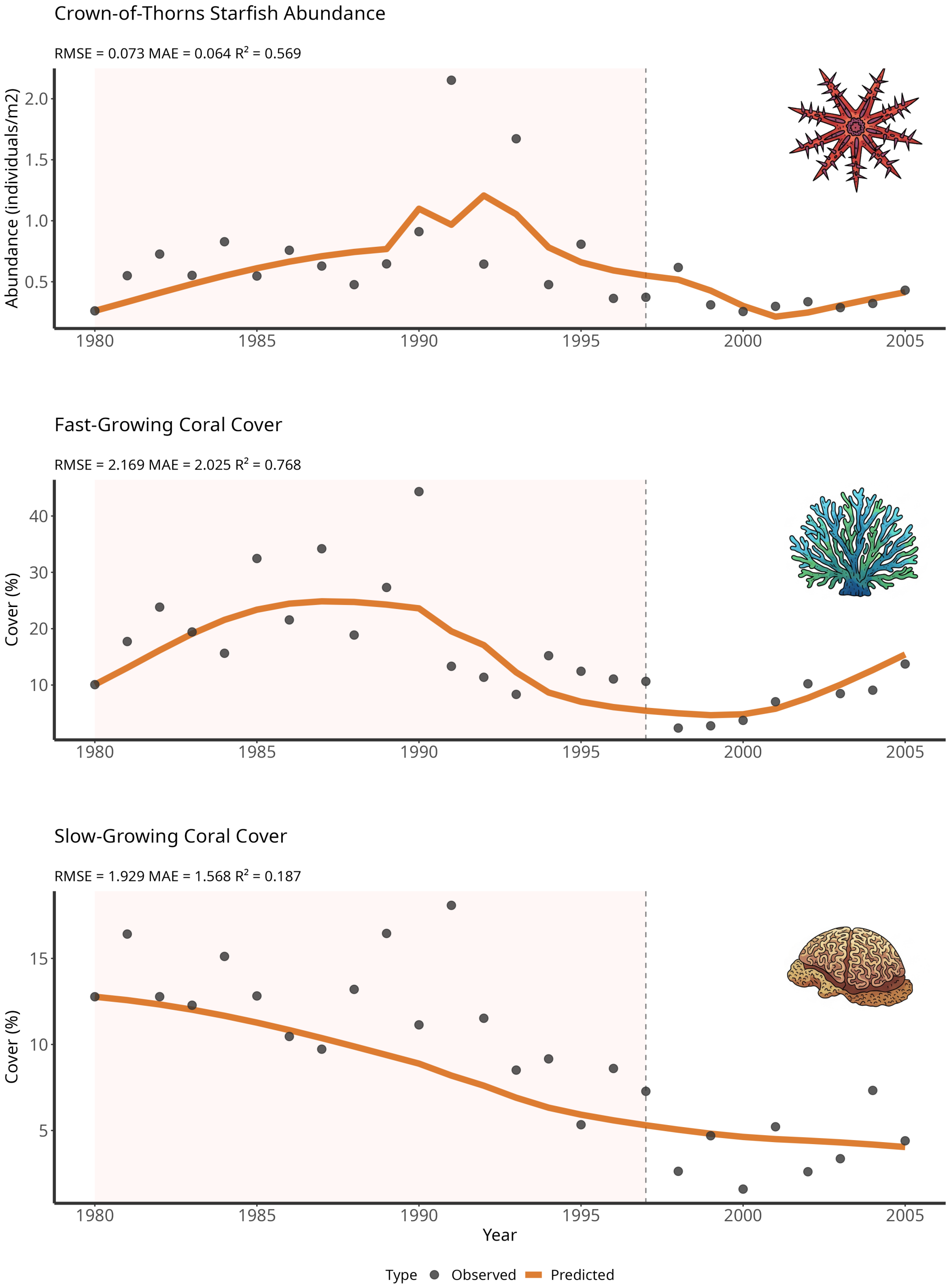

Figure 5: Time-series validation of the best-performing AI model (GPT 4.1) showing predictions against observed data. The model was trained on 70% of the time-series data (pink shaded region) and validated on the remaining unseen 30% of the time-series data (white region). Top: COTS abundance predictions showing some capture of population variability. Middle: Fast-growing coral cover predictions demonstrating tracking of recovering . 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 AIME 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 AIME successfully generating models that approached the predictive performance of human expert models, albeit with less consistency across all variables. While the AI-derived models captured key ecological dynamics, they did so using structurally simpler formulations, and their performance varied depending on the specific ecosystem component being modeled.

Notably, the AI-generated models achieved predictive performance that was broadly similar to that of the human expert model, despite substantial structural differences. These results suggest that, under certain conditions, AI-driven approaches may approximate expert-level outcomes, though further validation and refinement are needed to ensure robustness and ecological fidelity (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 and input from domain experts, while AI models achieved similar performance with more parsimonious formulations. This is consistent with the coexistence of multiple numerically valid representations of the same ecological system where each offers different insights into the underlying mechanisms[33], 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.

Importantly, this flexibility also opens the door to hybrid approaches that combine the strengths of both expert-driven and AI-generated models. By explicitly prompting the AI to include particular ecological processes, such as age structure, prey-switching behavior, or nutrient mixing, researchers can guide model structure to reflect known system dynamics or stakeholder priorities. This capability enhances the utility of AI-generated models for hypothesis testing, scenario exploration, and applied decision-making, particularly in contexts where certain mechanisms are known to be ecologically or socially important. 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 new 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[20], have shown high accuracy in direct forecasting of environmental variables[8,17]. 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[34]. 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[8] and root mean squared errors reduced by up to 52% compared to traditional methods[17]. While our evolved models may not always match these pure prediction accuracies, they offer advantages in interpretability, scientific insight and potential for strategic and tactical applications.

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 mechanistic understanding is as important as predictive accuracy. The interpretability of AI-evolved models enables users 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[7,27]. 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. Crucially, this process is not solely computational: models are refined through expert collaboration, which synthesizes domain knowledge and builds confidence and legitimacy in the resulting models. This step is essential for filtering out plausible but ultimately incorrect mechanisms that may arise from AI-driven parsimony. For example, some temperature-related functions affecting COTS in the AI-derived models lack empirical support and thus their inclusion would likely be removed during expert-led refinement. While such mechanisms may be ecologically plausible, their inclusion without validation risks misleading management decisions, especially under extreme conditions. Thus, expert review acts as an important filter, ensuring that models are not only interpretable, but also credible and useful for decision-making.

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, LLM temperature setting, 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 and Figure 9) reveals trade-offs between model choice and rate of improvement. While the o3-mini and o4-mini configurations were consistently able to iteratively improve, the Sonnet models and GPT 4.1 model were often 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[35–37]

Implications for Ecosystem-Based Fisheries Management

The successful application of AIME to COTS populations on the Great Barrier Reef demonstrates its potential for use in developing plausible models that can provie insights on pressing ecosystem problems relevant to ecosystem based management. While more development and experience is needed to ascertain AI-derived models directly into decision making contexts, there is certainly sufficient evidence that they can be used as a means of hypothesis generation and for the development of plausible models. This radically speeds up model buildng, bringing the process more into line with timeframes pertinent to pressing tactical management questions. The framework's capacity to capture both short-term outbreak dynamics and longer-term ecosystem changes provides scientists supporting 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.

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, such as in biosecurity emergencies, where management actions depend on rapid model development and deployment. AIME'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.

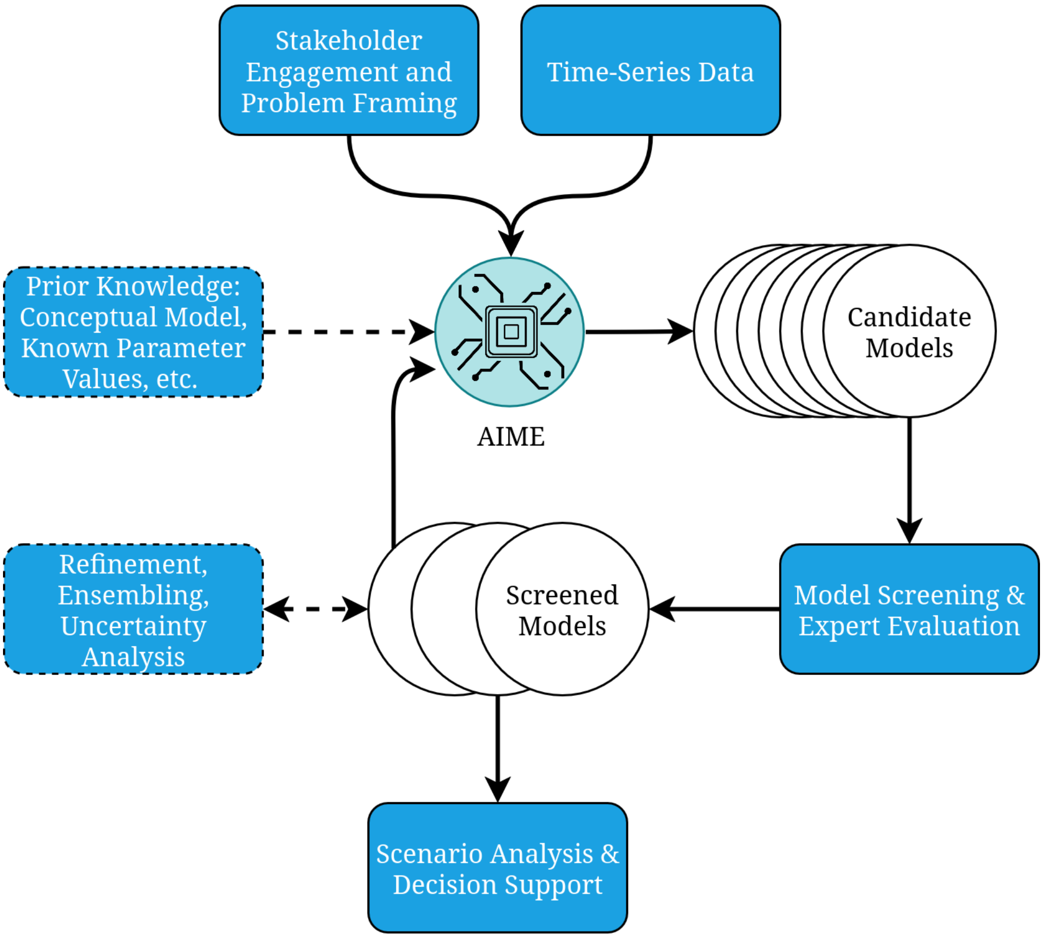

Figure 6: The AIME framework workflow integrating human expertise with AI-driven model development. The diagram illustrates how stakeholder engagement, time-series data, and prior ecological knowledge inform the AIME process, which generates candidate models that can be evaluated and refined by human experts, ultimately supporting ecosystem management decisions.

To ensure safe and effective integration of AIME into ecosystem-based fisheries management, we recommend a set of best practices and propose that AIME would work best embedded within a human-driven workflow (Figure 6).

1. **Stakeholder Engagement** Begin with stakeholder engagement to inform problem framing and define the ecological questions AIME will address. Early involvement ensures models are relevant to management needs and incorporate local knowledge.

2. **Expert Review** All AI-generated models should be reviewed by domain experts to validate ecological plausibility and ensure alignment with management objectives.

3. **Complementary Use** Use AIME as a complementary tool that supports - rather than replaces - traditional modelling workflows, especially for rapid prototyping or exploring alternative hypotheses.

4. **Model Transparency** Maintain transparency through clear documentation of equations, parameters, and assumptions to ensure traceability and reproducibility.

5. **Parameter Assessment** Critically assess parameter values sourced from literature. Fix well-established values where appropriate to reduce uncertainty.

6. **Rigorous Validation** Conduct thorough validation, including cross-validation and out-of-sample testing, before applying models to decision-making.

7. **Training and Capacity Building** Provide training to ensure managers and researchers can interpret and apply AI-generated models responsibly.

In conclusion, AIME represents an 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.

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. All of the code and data underpinning this study are available at the Github repository: `https://github.com/s-spillias/EMs-with-LLMs`.

AIME Framework

Model Generation and Improvement

AIME uses LLMs to write and modify computer code through Aider[38], 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 S3.1 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[31].

During non-initial steps, 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 S3.3).

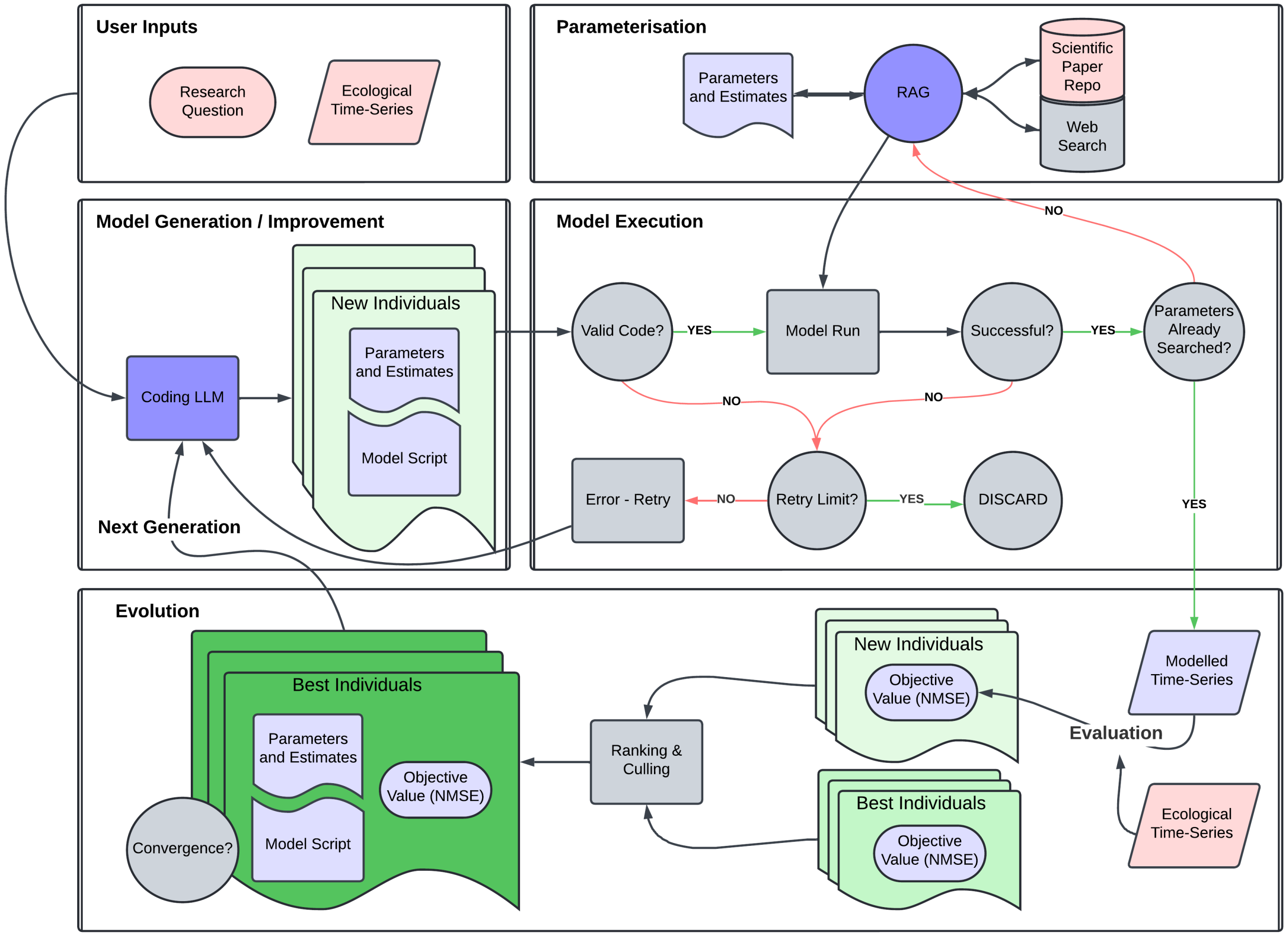

Figure 7: Conceptual diagram of the automated ecological modelling framework, AIME. The workflow consists of four main components: (1) User Inputs, where research questions and ecological time-series data are provided; (2) Parameterisation, utilizing RAG-enhanced literature search to estimate parameter 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.

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[26,39], the system implements a Retrieval-Augmented Generation (RAG) architecture to search scientific literature (see Section S2 for detailed RAG implementation). Without this initial estimation step, the system risks wasting time and computational resources searching for parameter values to populate equations that may be structurally or syntactically flawed. By first validating the integrity of the model through execution, the system ensures that subsequent efforts to refine parameter values are meaningful and efficient.

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 S1 for the complete collection used for the COTS case study) using ChromaDB[40] as a persistent vector store, with documents processed into semantic chunks to enable precise retrieval. Second, it queries the Semantic Scholar database[41]. Third, it performs general web searches through the Serper API[42] 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 S3.2). 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. Values derived from the literature were used to bound the feasible parameter space and inform initial parameter estimates. 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.

Models are executed through Template Model Builder, TMB[43], an approach which underpins several marine ecosystem modelling frameworks[44–46]. TMB uses automatic differentiation techniques to efficiently estimate parameters and is capable of handling complex and non-linear optimisation problems. Optimisation priorities for AIME using TMB are specified in accordance with established practices[30,31]. 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 S3.4.

Model Evaluation

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

$$
\text{NMSE}_j = \begin{cases} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_{ij}-\hat{y}_{ij}}{\sigma_j}\right)^2 & \text{if } \sigma_j \neq 0 \\ \frac{1}{n}\sum_{i=1}^{n}(y_{ij}-\hat{y}_{ij})^2 & \text{if } \sigma_j = 0 \end{cases} \tag{1}
$$

where y_{ij} represents observed values for variable j at time i, $\hat{y}_{ij}$ represents corresponding model predictions, σ_j is the unbiased standard deviation of the observed values for variable j (calculated with $n-1$ denominator), and n is the number of observations. The final objective function value is the mean across all response variables:

$$
\text{Objective} = \frac{1}{m}\sum_{j=1}^{m}\text{NMSE}_j \tag{2}
$$

where m 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 simplicity 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.

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[28], whose dynamics are well-studied[47,48]. The complete system of equations is presented in Section S3.5 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-3.7 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 $\frac{N}{k_N+N} \cdot \frac{k_I}{k_I+P}$), 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. We divided the original equations into 8 terms and thus the highest possible ecological score would be 8, representing perfect agreement between the AIME-generated model and the original model equations. The complete evaluation prompt with detailed scoring criteria for each ecological characteristic is provided in Section S3.5. 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-3.7, 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.

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 (GPT4.1, o3-mini and o4-mini from OpenAI, Claude Sonnet 3.6 and 3.7 from Anthropic, 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.

The COTS model that we used as a human-derived benchmark for evaluating AIME's outputs was originally developed to specifically evaluate management interventions[49]. Subsequent model versions and variations have yielded insight into management under environmental perturbations[30,32], derivation of management thresholds[50,51] and their dynamic implementation[52]. Each application differed depending on the objectives and data availability, and how the models were resolved, which required human determinations as to what was included, how it was included, and how it linked with other system aspects where necessary. Simply put, human experts were required to link management objectives and available data to resolve the necessary system aspects for informing specific management actions.

Here we test the ability of the AI to develop a model that captures the dynamics of corals and COTS during a COTS outbreak. This not only required the AI to link the ecological dynamics but also interpret the life history characteristics of COTS to explain the observed data. 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

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 R^2 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.

Acknowledgements

SS was supported by an R+ Postdoctoral Fellowship.

Code and Data Availability

The code for the AIME framework is available in a public GitHub repository at https://github.com/s-spillias/EMs-with-LLMs. The repository includes all scripts necessary to reproduce the results presented in this paper, including the genetic algorithm implementation, model evaluation tools, and analysis scripts. The time series data used in the case studies are also available in the repository. The software is released under an MIT license.

Declaration on Generative AI Usage

During the preparation of this manuscript, we used Claude-3.5-Sonnet, a large language model, to assist with code documentation, manuscript formatting, and language editing. The scientific content, analyses, interpretations, and conclusions presented in this paper were developed and validated by the human authors.

Author Contributions

SS: Conceptualization, Methodology, Software, Data curation, Formal analysis, Writing - original draft, Writing - review & editing JR: Software, Methodology, Data curation, Writing - review & editing FB: Formal analysis, Supervision, Writing - review & editing BF: Formal analysis, Supervision, Writing - review & editing RT: Supervision, Writing - review & editing MG: Software, Methodology, Writing - review & editing SY: Software, Methodology, Writing - review & editing

Competing Interests

The authors declare no competing interests.

Data Availability

The datasets generated and analyzed during the current study are available in the GitHub repository https://github.com/s-spillias/EMs-with-LLMs. Additional data that support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] McCarthy, M. A. & Possingham, H. P. Active adaptive management for conservation. *Conservation Biology* **18**, 534–541 (2004).

[2] Holden, P. B. & Ellner, S. P. Prediction in ecology: a first-principles framework. *Ecological Monographs* **86**, 303–319 (2016).

[3] Dichmont, C. M. *et al.* From data rich to data-limited harvest strategies—does more data mean better management? *ICES Journal of Marine Science* **74**, 670–686 (2017).

[4] Holden, M. H. *et al.* Cost–benefit analysis of ecosystem modeling to support fisheries management. *Journal of Fish Biology* **104**, 1667–1674 (2024).

[5] Weiskopf, S. R. *et al.* Climate change effects on biodiversity, ecosystems, ecosystem services, and natural resource management in the united states. *Science of the Total Environment* **733**, 137782 (2020).

[6] Malhi, Y. *et al.* Climate change and ecosystems: threats, opportunities and solutions (2020).

[7] Spillias, S. *et al.* The Future of Artificial Intelligence in Ecosystem Modelling. *Manuscript in preparation* (2024).

[8] Morales-García, J., Llanes, A., Arcas-Túnez, F. & Terroso-Sáenz, F. Developing time series forecasting models with generative large language models. *ACM Transactions on Intelligent Systems and Technology* (2024).

[9] Huntingford, C., Nicoll, A. J., Klein, C. & Ahmad, J. A. Potential for Equation Discovery with AI in the Climate Sciences (2024).

[10] Floryan, D. & Graham, M. D. Data-driven discovery of intrinsic dynamics. *Nature Machine Intelligence* **4**, 1113–1120 (2022).

[11] Schaeffer, H. Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society A* **473**, 20160446 (2017).

[12] Chen, X., Soh, B. W., Ooi, Z. E. *et al.* Constructing custom thermodynamics using deep learning. *Nature Computational Science* **4**, 66–85 (2024).

[13] Karniadakis, G. E. *et al.* Physics-informed machine learning. *Nature Reviews Physics* **3**, 422–440 (2021).

[14] Zhang, X., Chowdhury, R. R., Gupta, R. K. & Shang, J. Large language models for time series: A survey. *arXiv preprint arXiv:2402.01801* (2024).

[15] Su, J. *et al.* Large language models for forecasting and anomaly detection: A systematic literature review. *arXiv preprint arXiv:2402.10350* (2024).

[16] Hassani, H. & Silva, E. S. Predictions from Generative Artificial Intelligence Models: Towards a New Benchmark in Forecasting Practice. *Information* **15**, 291 (2024).

[17] Gandhi, Y. *et al.* Generative AI-driven forecasting of oil production. *arXiv preprint arXiv:2409.16482* (2024).

[18] Bylund, D. & Johansson, V. ChatGPT for Time Series Predictions (2024).

[19] Cao, D. *et al.* Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948* (2023).

[20] Li, H. *et al.* LITE: Modeling Environmental Ecosystems with Multimodal Large Language Models. *arXiv preprint arXiv:2404.01165* (2024).

[21] Garza, A., Challu, C. & Mergenthaler-Canseco, M. Timegpt-1. *arXiv preprint arXiv:2310.03589* (2023).

[22] Xu, Y. *et al.* Artificial intelligence: A powerful paradigm for scientific research. *The Innovation* **2**, 100179 (2021).

[23] The Royal Society. Science in the age of AI: How artificial intelligence is changing the nature and method of scientific research. Report, The Royal Society (2024).

[24] Burger, B. *et al.* Autonomous chemical research with large language models. *Nature* **624**, 505–512 (2023).

[25] Wang, Z., Zhang, Y., Zhao, Y., Chen, Y. & Li, P. BioResearcher: Automating Biomedical Research via Large Language Models. *arXiv preprint arXiv:2412.09429* (2024).

[26] Spillias, S. *et al.* Evaluating generative ai to extract qualitative data from peer-reviewed documents (2024).

[27] Kramer, S., Cerrato, M., Džeroski, S. & King, R. Automated scientific discovery: from equation discovery to autonomous discovery systems. *arXiv preprint arXiv:2305.02251* (2023).

[28] Edwards, A. M. & Brindley, J. Zooplankton mortality and the dynamical behaviour of plankton population models. *Bulletin of mathematical biology* **61**, 303–339 (1999).

[29] Morello, E. *et al.* Model to manage and reduce crown-of-thorns starfish outbreaks. *Marine Ecology Progress Series* **512**, 167–183 (2014).

[30] Rogers, J. G. D. & Plagányi, E. E. Culling corallivores improves short-term coral recovery under bleaching scenarios. *Nature Communications* **13**, 2520 (2022).

[31] Plagányi, E. E. *et al.* Multispecies fisheries management and conservation: tactical applications using models of intermediate complexity. *Fish and Fisheries* **15**, 1–22 (2014).

[32] Condie, S. A. *et al.* Large-scale interventions may delay decline of the Great Barrier Reef. *Royal Society Open Science* **8**, 201296 (2021).

[33] Patterson, K. *et al.* Estimating uncertainty in fish stock assessment and forecasting. *Fish and fisheries* **2**, 125–157 (2001).

[34] Adams, M. P. *et al.* Model fit versus biological relevance: Evaluating photosynthesis-temperature models for three tropical seagrass species. *Scientific reports* **7**, 39930 (2017).

[35] Baker, C. M., Gordon, A. & Bode, M. Ensemble ecosystem modeling for predicting ecosystem response to predator reintroduction. *Conservation Biology* **31**, 376–384 (2017).

[36] Gårdmark, A. *et al.* Biological ensemble modeling to evaluate potential futures of living marine resources. *Ecological applications* **23**, 742–754 (2013).

[37] Vollert, S. A., Drovandi, C. & Adams, M. P. Unlocking ensemble ecosystem modelling for large and complex networks. *PLoS Computational Biology* **20**, e1011976 (2024).

[38] Gauthier, P. Aider: AI pair programming in your terminal (2024). URL https://github.com/aider-ai/aider.

[39] Keck, F., Broadbent, H. & Altermatt, F. Extracting massive ecological data on state and interactions of species using large language models. *bioRxiv* 2025–01 (2025).

[40] Chroma. Chroma - the ai-native open-source embedding database (2024). URL https://docs.trychroma.com/docs/overview/introduction. Accessed: 2024.

[41] Allen Institute for AI. Semantic scholar academic graph api (2024). URL https://api.semanticscholar.org/.

[42] Serper. Serper search api (2024). URL https://serper.dev/.

[43] Kristensen, K. TMB: General random effect model builder tool inspired by ADMB. *R package version* **1** (2014).

[44] Chasco, B., Burke, B., Crozier, L. & Zabel, R. Differential impacts of freshwater and marine covariates on wild and hatchery chinook salmon marine survival. *PLoS One* **16**, e0246659 (2021).

[45] Albertsen, C. M., Whoriskey, K., Yurkowski, D., Nielsen, A. & Flemming, J. M. Fast fitting of non-gaussian state-space models to animal movement data via template model builder (2015).

[46] Auger-Méthé, M. *et al.* Spatiotemporal modelling of marine movement data using template model builder (tmb). *Marine Ecology Progress Series* **565**, 237–249 (2017).

[47] Boschetti, F. Mapping the complexity of ecological models. *Ecological Complexity* **5**, 37–47 (2008).

[48] Boschetti, F. Detecting behaviours in ecological models. *Ecological Complexity* **7**, 76–85 (2010).

[49] Morello, E. B. *et al.* Model to manage and reduce crown-of-thorns starfish outbreaks. *Marine Ecology Progress Series* **512**, 167–183 (2014).

[50] Plagányi, É. E., Babcock, R. C., Rogers, J., Bonin, M. & Morello, E. B. Ecological analyses to inform management targets for the culling of crown-of-thorns starfish to prevent coral decline. *Coral Reefs* **39**, 1483–1499 (2020).

[51] Rogers, J. G., Plagányi, É. E., Blamey, L. K. & Desbiens, A. A. Validating effectiveness of crown-of-thorns starfish control thresholds to limit coral loss throughout the great barrier reef. *Coral Reefs* 1–16 (2024).

[52] Rogers, J. G., Plagányi, É. E., Babcock, R. C., Fletcher, C. S. & Westcott, D. A. Improving coral cover using an integrated pest management framework. *Ecological Applications* **33**, e2913 (2023).

[53] Plagányi, E. E. Models for an ecosystem approach to fisheries (2007).

[54] Collie, J. S. *et al.* Ecosystem models for fisheries management: finding the sweet spot. *Fish and Fisheries* **17**, 101–125 (2016).

[55] Pratchett, M. *et al.* Thirty Years of Research on Crown-of-Thorns Starfish (1986–2016): Scientific Advances and Emerging Opportunities. *Diversity* **9**, 41 (2017).

[56] Blamey, L. K., Plagányi, E. E. & Branch, G. M. Was overfishing of predatory fish responsible for a lobster-induced regime shift in the Benguela? *Ecological Modelling* **273**, 140–150 (2014).

[57] Plagányi, E. *et al.* Ecosystem modelling provides clues to understanding ecological tipping points. *Marine Ecology Progress Series* **512**, 99–113 (2014).

[58] Punt, A. E. *et al.* Exploring the implications of the harvest control rule for Pacific sardine, accounting for predator dynamics: A MICE model. *Ecological Modelling* **337**, 79–95 (2016).

[59] Essington, T. E. & Plagányi, E. E. Pitfalls and guidelines for "recycling" models for ecosystem-based fisheries management: evaluating model suitability for forage fish fisheries. *ICES Journal of Marine Science* **71**, 118–127 (2014).

[60] Gamble, R. J. & Link, J. S. Analyzing the tradeoffs among ecological and fishing effects on an example fish community: A multispecies (fisheries) production model. *Ecological Modelling* **220**, 2570–2582 (2009).

[61] Hadley, S., Wild-Allen, K., Johnson, C. & Macleod, C. Modeling macroalgae growth and nutrient dynamics for integrated multi-trophic aquaculture. *Journal of Applied Phycology* **27**, 901–916 (2015).

[62] Oca, J., Cremades, J., Jiménez, P., Pintado, J. & Masaló, I. Culture of the seaweed Ulva ohnoi integrated in a Solea senegalensis recirculating

system: influence of light and biomass stocking density on macroalgae productivity. *Journal of Applied Phycology* **31**, 2461–2467 (2019).

[63] Spillias, S., Cottrell, R. S., Layton, C., O'Brien, K. R. & McDonald-Madden, E. Having our kelp and eating it too: Minimizing trade-offs from seaweed farming. *Journal of Cleaner Production* **448**, 141150 (2024).

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

S1 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: [53] established foundational principles for ecosystem approaches to fisheries, while [31] introduced Models of Intermediate Complexity for Ecosystem assessments (MICE). [54] explored optimal model complexity levels.

- COTS Management and Ecology: [55] provided a comprehensive thirty-year review of COTS research. [29] developed models for COTS outbreak management, while [30] analyzed corallivore culling impacts under bleaching scenarios.

- Ecological Regime Shifts: [56] investigated predator-driven regime shifts in marine ecosystems. [57] provided insights into ecological tipping points through ecosystem modeling.

- Management Interventions: [32] examined large-scale interventions on the Great Barrier Reef. [58] explored harvest control implications using MICE models.

- Model Application Guidelines: [59] provided critical guidelines for adapting ecosystem models to new applications. [60] demonstrated multispecies production model applications for analyzing ecological and fishing effects.

- Integrated Systems: [61] and [62] explored integrated multi-trophic aquaculture modeling, providing insights into coupled biological systems. [63] 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.

S2 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.

S3 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.

S3.1 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
```

S3.2 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.
```

S3.3 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.
```

S3.4 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
```

S3.5 NPZ Case Study - Recovering Equations

The model implementation can be compared to the original NPZ equations from[28]:

$$
\begin{aligned}
\frac{dN}{dt} &= \underbrace{-\frac{V_m N P}{k_s + N}}_{\text{nutrient uptake}} \underbrace{+\gamma(1-\alpha)\frac{gP^2 Z}{k_g + P^2} + \mu_P P + \mu_Z Z^2}_{\text{recycling}} + \underbrace{S(N_0 - N)}_{\text{mixing}} \\
\frac{dP}{dt} &= \underbrace{\frac{V_m N P}{k_s + N}}_{\text{growth}} \underbrace{- \frac{gP^2 Z}{k_g + P^2} - \mu_P P - SP}_{\text{losses}} \\
\frac{dZ}{dt} &= \underbrace{\alpha \frac{gP^2 Z}{k_g + P^2} - \mu_Z Z^2 - SZ}_{\text{growth and mortality}}
\end{aligned}
$$

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.

S4 NPZ Validation

The NPZ validation study evaluated AIME's ability to recover known ecological relationships from synthetic data. Figure 8 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.

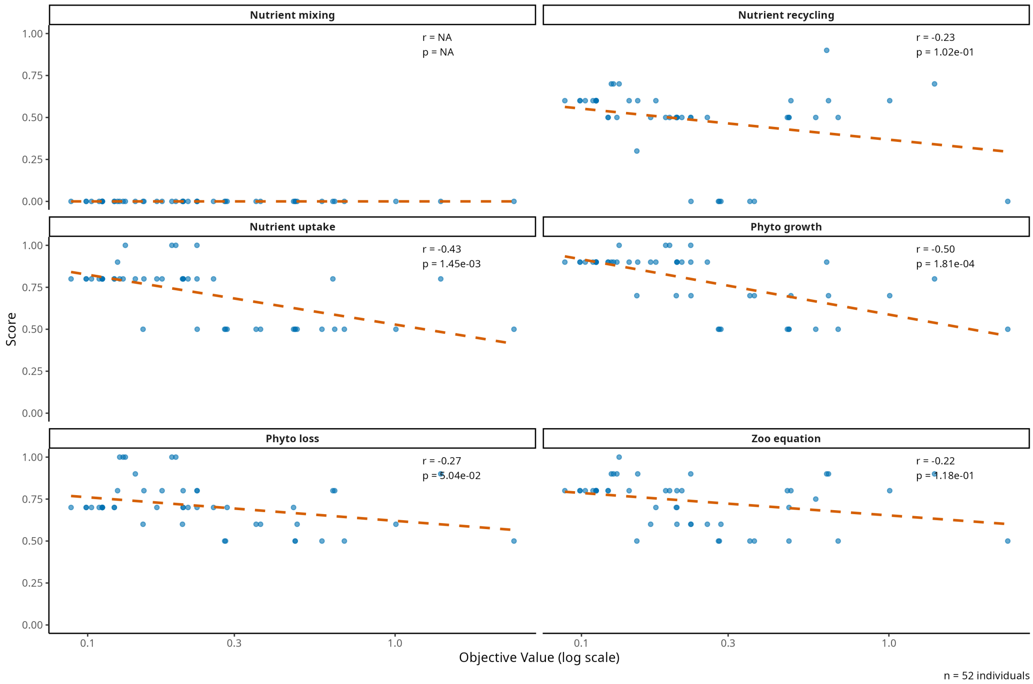

Figure 8: Relationship between ecological accuracy scores and model performance for each NPZ model characteristic. Each panel shows how well models recovered a specific ecological mechanism (score from 0-1) versus their predictive accuracy (objective value). Lower objective values indicate better model fit. Two-sided Pearson's product-moment correlation coefficients (r) and their associated p-values are shown for each characteristic.

S4.1 Best Performing NPZ Model

This model achieved an objective value of 0.0883.

S4.1.1 Model Description

The following model represents our framework's attempt to recover the NPZ dynamics from [28]. The model aims to capture three key components:

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

S4.1.2 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.
```

S4.1.3 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;
}
```

S4.1.4 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
    }
  ]
}
```

S5 CoTS Model Convergence

S5.1 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 9 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.

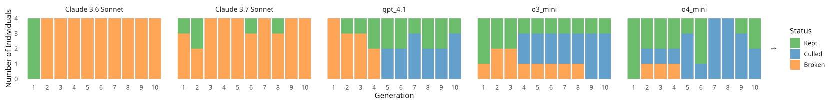

Figure 9: Evolution of model performance during the genetic algorithm optimization process. Each generation represents an iteration of model development, where models are evaluated and classified into three categories: the best performers according to the NMSE objective value (kept, green), those that 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.

S5.2 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 10).

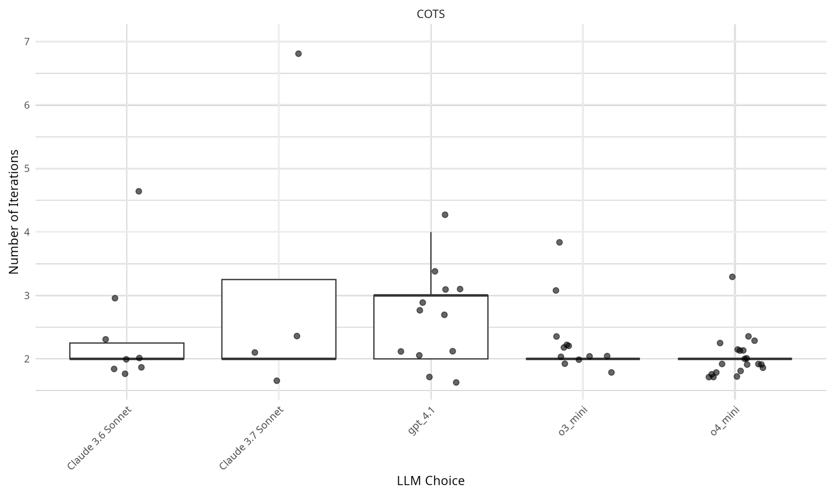

Figure 10: Distribution of iteration counts for successful model instances by LLM configuration. The boxplot shows the number of iterations required for convergence, excluding cases that reached maximum iterations or remained numerically unstable.

S6 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.

S6.1 Key Parameter Comparison

Table 1 presents a detailed comparison of key parameters across the five best-performing models and the human-generated model. These parameters represent fundamental ecological processes and reveal different modelling approaches to COTS-coral dynamics.

Table 1: Comparison of key parameters across best-performing models

Parameter	Human Model	o3 mini	Claude Sonnet 3.6	Claude Sonnet 3.7	o4 mini	gpt 4.1
COTS growth rate (yr^{-1})	Beverton-Holt (h=0.5)	exp(log_growth_rate)	0.8	0.8	0.5	0.5
COTS mortality (yr^{-1})	Mcots = 2.3	exp(log_decline_rate)	–	0.4	0.3	0.37
COTS carrying capacity	Derived from R0=1.0	–	2.0	2.5	50	0.61
Slow coral growth (yr^{-1})	rm = 0.1	0.1 (fixed)	0.2	0.1	0.05 (fixed)	0.37
Fast coral growth (yr^{-1})	rf = 0.5	0.2 (fixed)	0.4	0.3	0.1 (fixed)	0.61
Coral carrying capacity	K = 3000 (shared)	–	0.8	K_slow = 30, K_fast = 50	–	K_slow = 20.1, K_fast = 33.1
Fast coral optimal temp (°C)	SST0_f = 26	–	–	–	–	–
Slow coral optimal temp (°C)	SST0_m = 27	–	–	–	–	–
COTS optimal temp (°C)	Implicit	–	28	28	–	–
Attack rate (fast coral)	p1f = 0.15	0.4	0.1	0.2	0.05	0.14
Attack rate (slow coral)	p1m = 0.06	0.6	0.05	0.05	0.05	0.08
Predation switching	switchSlope = 5	–	–	–	–	–
Functional response	Sigmoid switching	Logistic with quadratic adjustment	Type II	Type II	Type III	Type II

S6.2 Model Structure Comparison

Table 2 presents a detailed comparison of the key equations used in each model, highlighting the different mathematical approaches to representing COTS-coral dynamics.

Table 2: Comparison of key equations across models

Model	Key Equations
Human Model	**COTS dynamics:** Age-structured model with three age classes (0, 1, 2+) $N(yr+1,1) = N(yr,0) \cdot \exp(-1 \cdot M\_CoTS\_age(0))$ $N(yr+1,2) = N(yr,1) \cdot \exp(-f \cdot M\_CoTS\_age(1)) + N(yr,2) \cdot \exp(-f \cdot M\_CoTS\_age(2))$ $Rcots(yr+1) = \frac{\alpha \cdot (N(yr+1,2)/Kots\_sp)}{\beta + (N(yr+1,2)/Kots\_sp)}$ $N(yr+1,0) = (Rcots(yr+1) + Imm\_CoTS) \cdot \exp(Imm\_res(yr+1) + \sigma^2_{CoTS}/2)$ **Coral dynamics:** $Cf(yr+1) = Cf(yr) \cdot (1.0 + \rho_{SST\_F} \cdot rf \cdot (1 - (Cf(yr) + Cm(yr))/K)) - Qf - M\_ble\_f$ $Cm(yr+1) = Cm(yr) \cdot (1.0 + \rho_{SST\_M} \cdot rm \cdot (1 - (Cf(yr) + Cm(yr))/K)) - Qm - M\_ble\_m$ **Predation:** $\rho = \exp(-switchSlope \cdot Cf(yr)/K)$ $Qf = Cf(yr) \cdot (1.0 - \rho) \cdot p1f \cdot \frac{N(yr,1)+N(yr,2)}{1.0+\exp(-(N(yr,1)+N(yr,2))/p2f)}$ $Qm = Cm(yr) \cdot \rho \cdot p1m \cdot \frac{N(yr,1)+N(yr,2)}{1.0+\exp(-(N(yr,1)+N(yr,2))/p2m)}$ **Temperature effects:** $\rho_{SST\_F} = \exp(-\frac{(SST-SST0\_f)^2}{2 \cdot SST\_sig\_f^2})$ $\rho_{SST\_M} = \exp(-\frac{(SST-SST0\_m)^2}{2 \cdot SST\_sig\_m^2})$ $M\_ble\_f = Cf(yr) \cdot \frac{1.0}{1.0+\exp(-Eta\_f \cdot (SST - M\_SST50\_f))}$ $M\_ble\_m = Cm(yr) \cdot \frac{1.0}{1.0+\exp(-Eta\_m \cdot (SST - M\_SST50\_m))}$

Continued on next page

Table 2 – continued from previous page

Model	Key Equations
o3 mini	**COTS dynamics:** $logistic\_factor = \frac{1}{1+\exp(-outbreak\_steepness\cdot(resource\_limitation-threshold))}$ $quadratic\_adjustment = \begin{cases} 1 + poly\_coeff \cdot (resource\_limitation - threshold) \\ 1 \end{cases}$ $outbreak\_factor = logistic\_factor \cdot quadratic\_adjustment$ $temperature\_factor = 1 + effect\_sst \cdot sst\_dat(t-1)$ $cots\_pred[t] = cots\_pred[t-1] + (growth\_rate \cdot cots\_pred[t-1] \cdot outbreak\_factor \cdot temperature\_factor - decline\_rate \cdot cots\_pred[t-1]) \cdot dt$ **Coral dynamics:** $fast\_pred[t] = fast\_pred[t-1] + dt \cdot (fast\_growth\_rate \cdot fast\_pred[t-1] \cdot (1 - fast\_pred[t-1]/fast\_cap) - efficiency\_fast \cdot cots\_pred[t-1] \cdot fast\_pred[t-1])$ $mod\_eff\_slow = efficiency\_slow \cdot (1 + temp\_mod\_eff\_slow \cdot sst\_dat(t-1))$ $slow\_pred[t] = slow\_pred[t-1] + dt \cdot (slow\_growth\_rate \cdot slow\_pred[t-1] \cdot (1 - slow\_pred[t-1]/slow\_cap) - mod\_eff\_slow \cdot cots\_pred[t-1] \cdot slow\_pred[t-1])$
Claude Sonnet 3.6	**COTS dynamics:** $temp\_effect = \exp(-0.5 \cdot \frac{(sst\_dat(t-1)-temp\_opt)^2}{temp\_range^2})$ $resource\_limit = \frac{total\_coral}{total\_coral+\epsilon}$ $recruitment = cotsimm\_dat(t-1) \cdot temp\_effect$ $cots\_pred(t) = cots\_pred(t-1) \cdot (1 + r\_cots \cdot resource\_limit \cdot (1 - \frac{cots\_pred(t-1)}{K\_cots})) + recruitment$ **Coral dynamics:** $coral\_space = \max(0, 1 - \frac{fast\_pred(t-1)+slow\_pred(t-1)}{100\cdot coral\_limit})$ $fast\_growth = r\_fast \cdot fast\_pred(t-1) \cdot coral\_space$ $fast\_pred\_loss = grazing\_fast \cdot cots\_pred(t-1) \cdot fast\_pred(t-1)$ $fast\_pred(t) = fast\_pred(t-1) + fast\_growth - fast\_pred\_loss$ $slow\_growth = r\_slow \cdot slow\_pred(t-1) \cdot coral\_space$ $slow\_pred\_loss = grazing\_slow \cdot cots\_pred(t-1) \cdot slow\_pred(t-1)$ $slow\_pred(t) = slow\_pred(t-1) + slow\_growth - slow\_pred\_loss$

Continued on next page

Table 2 – continued from previous page

Model	Key Equations
Claude Sonnet 3.7	**COTS dynamics:** $temp\_effect = \exp(-0.5 \cdot \frac{(sst - temp\_opt)^2}{temp\_width^2})$ $pred\_fast = \frac{a\_fast \cdot fast\_t0 \cdot cots\_t0}{1.0 + a\_fast \cdot h\_fast \cdot fast\_t0 + a\_slow \cdot h\_slow \cdot slow\_t0 + \epsilon}$ $pred\_slow = \frac{a\_slow \cdot slow\_t0 \cdot cots\_t0}{1.0 + a\_fast \cdot h\_fast \cdot fast\_t0 + a\_slow \cdot h\_slow \cdot slow\_t0 + \epsilon}$ $bleach\_effect = \frac{1.0}{1.0 + \exp(-2.0 \cdot (sst - bleach\_threshold))}$ $cots\_growth = r\_cots \cdot cots\_t0 \cdot (1.0 - \frac{cots\_t0}{K\_cots}) \cdot temp\_effect$ $imm\_term = \frac{imm\_effect \cdot cotsimm}{1.0 + cotsimm + \epsilon}$ $food\_limitation = m\_cots \cdot (1.0 + \frac{1.0}{fast\_t0 + slow\_t0 + \epsilon})$ $cots\_pred(t) = cots\_t0 + cots\_growth - food\_limitation \cdot cots\_t0 + imm\_term$ **Coral dynamics:** $fast\_growth = r\_fast \cdot fast\_t0 \cdot (1.0 - \frac{fast\_t0 + competition \cdot slow\_t0}{K\_fast})$ $fast\_bleaching = bleach\_mortality\_fast \cdot bleach\_effect \cdot fast\_t0$ $fast\_pred(t) = fast\_t0 + fast\_growth - pred\_fast - fast\_bleaching$ $slow\_growth = r\_slow \cdot slow\_t0 \cdot (1.0 - \frac{slow\_t0 + competition \cdot fast\_t0}{K\_slow})$ $slow\_bleaching = bleach\_mortality\_slow \cdot bleach\_effect \cdot slow\_t0$ $slow\_pred(t) = slow\_t0 + slow\_growth - pred\_slow - slow\_bleaching$

Continued on next page

Table 2 – continued from previous page

Model	Key Equations
o4 mini	**COTS dynamics:** $coral\_availability = \frac{fast\_pred[t-1]+slow\_pred[t-1]}{200}$ $resource\_factor = \frac{coral\_availability+coral\_saturation\_coefficient\cdot coral\_availability^2}{0.5+coral\_availability+coral\_saturation\_coefficient\cdot coral\_availability^2}$ $growth = growth\_rate\_cots\cdot cots\_pred[t-1]\cdot(1-\frac{cots\_pred[t-1]}{carrying\_capacity+\epsilon})\cdot (1+resource\_limitation\_strength\cdot(resource\_factor-0.5))$ $effective\_sharpness = outbreak\_sharpness \cdot environmental\_modifier \cdot (1 + extreme\_outbreak\_modifier \cdot (environmental\_modifier-1))$ $raw\_trigger = \frac{1}{1+\exp(-effective\_sharpness\cdot(cots\_pred[t-1]^{outbreak\_shape}+outbreak\_nonlinearity\cdot$ $outbreak\_trigger = raw\_trigger + outbreak\_hysteresis \cdot raw\_trigger\cdot(1-raw\_trigger)$ $decline = decay\_rate\_cots\cdot cots\_pred[t-1]^{outbreak\_decline\_exponent}\cdot outbreak\_trigger$ $cots\_pred[t] = cots\_pred[t-1]+growth-decline$ **Coral dynamics:** $fast\_pred[t] = fast\_pred[t-1]+0.1\cdot coral\_recovery\_modifier\cdot coral\_recovery\_environmental\_modifier\cdot(100-fast\_pred[t-1])\cdot(1-coral\_recovery\_inhibition\cdot\frac{cots\_pred[t-1]}{carrying\_capacity+\epsilon}) - \frac{cots\_pred[t-1]\cdot coral\_predation\_efficiency\cdot fast\_pred[t-1]\cdot(\frac{fast\_pred[t-1]}{fast\_pred[t-1]+predation\_scaler})^{predation\_effic}}{1+handling\_time\cdot fast\_pred[t-1]}$ $slow\_pred[t] = slow\_pred[t-1]+0.05\cdot coral\_recovery\_environmental\_modifier\cdot(100-slow\_pred[t-1])\cdot(1-coral\_recovery\_inhibition\cdot\frac{cots\_pred[t-1]}{carrying\_capacity+\epsilon}) - \frac{cots\_pred[t-1]\cdot coral\_predation\_efficiency\cdot slow\_pred[t-1]\cdot(\frac{slow\_pred[t-1]}{slow\_pred[t-1]+predation\_scaler})^{predation\_effic}}{1+handling\_time\cdot slow\_pred[t-1]}$

Continued on next page

Table 2 – continued from previous page

Model	Key Equations
gpt 4.1	**COTS dynamics:** $coral\_effect = \frac{fast\_prev \cdot e\_fast + slow\_prev \cdot e\_slow}{K\_fast \cdot e\_fast + K\_slow \cdot e\_slow + \epsilon}$ $sst\_effect = 1.0 + theta\_sst \cdot (sst\_dat(t) - 27.0)$ $immig\_effect = immig\_scale \cdot cotsimm\_dat(t)$ $outbreak\_boost = 1.0 + phi\_outbreak \cdot (coral\_effect - 0.5)$ $cots\_growth = r\_cots \cdot cots\_prev \cdot (1.0 - \frac{cots\_prev}{K\_cots + \epsilon}) \cdot coral\_effect \cdot sst\_effect \cdot outbreak\_boost$ $cots\_mortality = m\_cots \cdot cots\_prev$ $cots\_next = cots\_prev + cots\_growth - cots\_mortality + immig\_effect$ **Coral dynamics:** $pred\_fast = \frac{\alpha\_fast \cdot cots\_prev \cdot fast\_prev}{fast\_prev + slow\_prev + \epsilon}$ $pred\_slow = \frac{\alpha\_slow \cdot cots\_prev \cdot slow\_prev}{fast\_prev + slow\_prev + \epsilon}$ $fast\_growth = r\_fast \cdot fast\_prev \cdot (1.0 - \frac{fast\_prev}{K\_fast + \epsilon})$ $fast\_mortality = m\_fast \cdot fast\_prev$ $fast\_next = fast\_prev + fast\_growth - pred\_fast - fast\_mortality$ $slow\_growth = r\_slow \cdot slow\_prev \cdot (1.0 - \frac{slow\_prev}{K\_slow + \epsilon})$ $slow\_mortality = m\_slow \cdot slow\_prev$ $slow\_next = slow\_prev + slow\_growth - pred\_slow - slow\_mortality$

S6.3 Detailed Ecological Mechanisms

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

S6.3.1 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 Sonnet 3.6: Uses a standard Gaussian temperature effect (temp_opt = 28°C) similar to the human model but without the explicit bleaching threshold.

Claude Sonnet 3.7: Implements a Gaussian temperature response with temp_opt = 28°C and temp_width = 2°C, affecting COTS recruitment. Also includes a bleaching effect with a threshold of 30°C.

o4 mini: Does not include explicit temperature dependency for COTS in its core equations, focusing instead on resource limitation and outbreak dynamics.

gpt 4.1: Implements a linear temperature effect where SST modifies growth (centered at 27°C) through the parameter theta_sst.

S6.3.2 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 direct predation with efficiency factors of 0.4 for fast coral and 0.6 for slow coral, with temperature modifying the predation efficiency on slow coral.

Claude Sonnet 3.6: Uses a simple Type II functional response with grazing rates of 0.1 for fast coral and 0.05 for slow coral, creating a saturating predation effect at high prey densities.

Claude Sonnet 3.7: Employs a Holling Type II functional response with attack rates of 0.2 for fast coral and 0.05 for slow coral, with handling times for both coral types.

o4 mini: Implements a Type III functional response with predation efficiency of 0.05, creating a sigmoidal functional response that reduces predation at low prey densities.

gpt 4.1: Uses a Type II functional response with attack rates of 0.14 for fast coral and 0.08 for slow coral, with predation partitioned by coral type.

S6.3.3 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. The models use various forms of logistic growth (o3 mini, Claude Sonnet 3.7, o4 mini, gpt 4.1) or temperature-modified reproduction functions (Claude Sonnet 3.6, Claude Sonnet 3.7, gpt 4.1).

S6.4 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. The gpt 4.1 model also implements a form of prey partitioning based on relative coral abundance.

Temperature effects: The human model implements temperature effects through two distinct mechanisms: (1) Gaussian functions modifying coral growth rates, similar to Claude Sonnet 3.6 and Claude Sonnet 3.7, and (2) a logistic bleaching mortality function with temperature thresholds, which is also implemented in Claude Sonnet 3.7 with its bleach_threshold parameter of 30°C.

Parameter differences: The human model uses different parameterization approaches, including:

- Direct parameterization of carrying capacity (K = 3000) rather than log-transformed values used in Claude Sonnet 3.6, Claude Sonnet 3.7, and gpt 4.1
- Separate parameters for predation intensity (p1f = 0.15, p1m = 0.06) and density-dependence (p2f, p2m)
- Explicit bleaching threshold parameters (M_SST50_f, M_SST50_m) compared to the single bleach_threshold in Claude Sonnet 3.7
- Age-dependent mortality components for COTS (Mcots = 2.3) compared to simpler mortality formulations in the AI models (e.g., m_cots = 0.4 in Claude Sonnet 3.7, 0.3 in o4 mini, and 0.37 in gpt 4.1)
- Explicit optimal temperatures for both coral types (SST0_f = 26°C, SST0_m = 27°C) which are not specified in the AI models

S6.5 Carrying Capacity and Growth Rate Comparison

The models show variation in their parameterization of carrying capacity and growth rates:

COTS carrying capacity: Values range from 0.61 individuals/m^2 (gpt 4.1) to 50 individuals/m^2 (o4 mini), with Claude Sonnet 3.6 at 2.0 and Claude Sonnet 3.7 at 2.5. This order-of-magnitude difference reflects fundamentally different assumptions about ecosystem capacity.

COTS growth rate: More consistency is observed in growth rates, with values of 0.8 per year (Claude Sonnet 3.6 and Claude Sonnet 3.7), 0.5 per year (o4 mini and gpt 4.1), compared to the Beverton-Holt approach in the human model.

Coral growth rates: The models show variation in coral growth parameters, with fast coral growth rates ranging from 0.1 per year (o4 mini) to 0.61 per year (gpt 4.1), and slow coral growth rates from 0.05 per year (o4 mini) to 0.37 per year (gpt 4.1).

Coral carrying capacity: The models use different approaches to coral carrying capacity, from the shared K = 3000 in the human model to separate values for fast and slow coral in Claude Sonnet 3.7 (K_fast = 50, K_slow = 30) and gpt 4.1 (K_fast = 33.1, K_slow = 20.1).

S7 Best Performing Models for CoTS Case Study

This section presents the 5 best performing models from different configurations for the Crown of Thorns Starfish case study.

S7.1 o3 mini Model

This model achieved an objective value of 0.5606.

S7.1.1 Model Intention

```
Model Assessment and Ecological Improvement Rationale:

Assessment:
- The current model captures boom-bust cycles but the logistic
   outbreak trigger appears too smooth.
- Residual analysis indicates that the model underestimates rapid
   outbreak escalation when coral resource availability is high.
- While the resource limitation function accounts for saturation, it
     may not fully capture synergistic effects at high coral cover.

Chosen Improvement:
- Introduce a quadratic adjustment factor to the outbreak trigger.
   When the resource availability (reflected by coral coverage)
   exceeds the threshold, the quadratic term amplifies the outbreak
   growth.
- This additional term (scaled by the new parameter poly_coeff) is
   ecologically justified by the abrupt and intense outbreak
   episodes observed in the Great Barrier Reef.
```

S7.1.2 Model Implementation

```
#include <TMB.hpp>

// 1. Data:
//    - Year: Time steps from the data file.
//    - sst_dat: Sea Surface Temperature data (°C).
//    - cotsimm_dat: Crown-of-thorns larval immigration rate (
   individuals/m2/year).
//    - cots_dat: Adult COTS abundance (individuals/m2).
//    - fast_dat: Fast-growing coral cover (Acropora spp.) in %.
//    - slow_dat: Slow-growing coral cover (Faviidae spp. and
   Porities spp.) in %.
//
// 2. Parameters and equations:
//    (1) COTS outbreak dynamics:
//        cots_pred[t] = cots_pred[t-1] +
//          [ growth_rate * cots_pred[t-1] * ( (fast_dat[t-1]+
   slow_dat[t-1])/(fast_dat[t-1]+slow_dat[t-1]+saturation) )
//            - decline_rate * cots_pred[t-1] ] * dt
//    (2) Environmental modification through sea surface temperature
     is embedded in the outbreak growth.
//    (3) Smooth transitions and small constants (e.g., 1e-8) are
   used to avoid division by zero.
//    (4) Only previous time step values are used in predictions to
   avoid data leakage.
//
// 3. Likelihood:
//    - Observations (cots_dat) are assumed to follow a lognormal
   distribution around the predictions.
//    - A fixed minimum standard deviation is used for numerical
   stability.
template<class Type>
Type objective_function<Type>::operator() ()
{
  // Data inputs from file
  DATA_VECTOR(Year);                  // Time (years)
  DATA_VECTOR(sst_dat);               // Sea Surface Temperature (°C)
  DATA_VECTOR(cotsimm_dat);           // COTS larval immigration rate
    (individuals/m2/year)
  DATA_VECTOR(cots_dat);              // Adult COTS abundance (
    individuals/m2)
```

```
DATA_VECTOR(fast_dat);           // Fast-growing coral cover (
  Acropora spp.) in %
DATA_VECTOR(slow_dat);           // Slow-growing coral cover (
  Faviidae spp. and Porities spp.) in %

int n = Year.size();             // Number of time steps

// Parameters (log-scale parameters ensure positivity)
PARAMETER(log_growth_rate);      // Log of intrinsic outbreak
  growth rate (log(year^-1))
PARAMETER(log_decline_rate);     // Log of outbreak decline rate
  (log(year^-1))
PARAMETER(log_threshold);        // Log of threshold resource
  level triggering outbreak (log(units))
PARAMETER(efficiency_fast);      // Efficiency factor for
  predation on fast-growing corals (unitless)
PARAMETER(efficiency_slow);      // Efficiency factor for
  predation on slow-growing corals (unitless)
PARAMETER(temp_mod_eff_slow);    // Temperature modifier for
  predation efficiency on slow-growing coral (unitless)
PARAMETER(effect_sst);           // Effect of sea surface
  temperature on outbreak progression (per °C)
PARAMETER(log_saturation);       // Log of saturation constant
  for resource limitation (log(units))
PARAMETER(log_outbreak_steepness);  // Log of outbreak steepness
  coefficient controlling outbreak trigger sensitivity
PARAMETER(poly_coeff);           // Quadratic adjustment
  parameter for outbreak triggering

// Parameter transformations to ensure positivity where applicable
Type growth_rate = exp(log_growth_rate);    // Intrinsic growth
  rate (year^-1)
Type decline_rate = exp(log_decline_rate);    // Decline rate
  during bust (year^-1)
Type threshold    = exp(log_threshold);       // Threshold
  resource level (units)
Type saturation   = exp(log_saturation);      // Saturation
  constant (units)
Type outbreak_steepness = exp(log_outbreak_steepness);

// Likelihood accumulation
Type nll = 0.0;
```

```
// Predicted state vectors for adult COTS numbers and coral covers
vector<Type> cots_pred(n);
vector<Type> fast_pred(n);
vector<Type> slow_pred(n);
cots_pred[0] = cots_dat(0);   // Initialize COTS with first
 observed value
fast_pred[0] = fast_dat(0);    // Initialize fast-growing coral
 with first observed value
slow_pred[0] = slow_dat(0);    // Initialize slow-growing coral
 with first observed value

for(int t = 1; t < n; t++){
  // 1. Calculate total coral cover from previous time step with a
   small constant to prevent division by zero.
  Type coral_total = fast_dat(t-1) + slow_dat(t-1) + Type(1e-8);
  // 2. Resource limitation modeled as a saturating function.
  Type resource_limitation = coral_total / (coral_total +
 saturation);

  // 3. Time difference between measurements
  Type dt = Year(t) - Year(t-1);
  // Introduce environmental modification: Sea Surface Temperature
   effect modulates outbreak growth rate.
  Type temperature_factor = 1 + effect_sst * sst_dat(t-1);

  // 4. COTS outbreak dynamics with non-linear outbreak threshold:
  //    Growth is modified by a logistic outbreak factor that
 captures triggering of outbreak events when
  //    resource availability (reflected by resource_limitation)
 exceeds the threshold. The steepness of this trigger
  //    is controlled by outbreak_steepness. The
 temperature_factor further modulates the growth rate.
  {
    Type logistic_factor = 1 / (Type(1) + exp(-outbreak_steepness
 * (resource_limitation - threshold)));
    Type quadratic_adjustment = CppAD::CondExpGt(
        resource_limitation,
        threshold,
        Type(1) + poly_coeff * pow(resource_limitation - threshold
 , 2),
        Type(1)
    );
    Type outbreak_factor = logistic_factor * quadratic_adjustment;
```

```
  cots_pred[t] = cots_pred[t-1] +
                 (growth_rate * cots_pred[t-1] * outbreak_factor
 * temperature_factor - decline_rate * cots_pred[t-1]) * dt;
 }

 // 5. Ensure numerical stability by preventing negative
predictions:
 cots_pred[t] = CppAD::CondExpGt(cots_pred[t], Type(1e-8),
cots_pred[t], Type(1e-8));

 // 5a. Coral dynamics:
 // Fast-growing coral dynamics
 Type fast_growth_rate = Type(0.2);  // (year^-1), assumed
constant growth rate
 Type fast_cap = Type(100.0);        // (%) maximum coral cover
 fast_pred[t] = fast_pred[t-1] + dt * ( fast_growth_rate *
fast_pred[t-1] * (1 - fast_pred[t-1] / fast_cap)
                     - efficiency_fast * cots_pred[t-1] *
fast_pred[t-1] );

 // Slow-growing coral dynamics
 Type slow_growth_rate = Type(0.1);  // (year^-1), assumed
constant growth rate
 Type slow_cap = Type(100.0);        // (%) maximum coral cover
 Type mod_eff_slow = efficiency_slow * (1 + temp_mod_eff_slow *
sst_dat(t-1));
 slow_pred[t] = slow_pred[t-1] + dt * ( slow_growth_rate *
slow_pred[t-1] * (1 - slow_pred[t-1] / slow_cap)
                     - mod_eff_slow * cots_pred[t-1] * slow_pred
[t-1] );

 // 6. Likelihood contribution:
 //    Observations cots_dat are assumed lognormally distributed
about predictions.
 //    A fixed standard deviation sigma = 0.1 is used for
numerical robustness.
 Type sigma = 0.1;
 nll -= dnorm(log(cots_dat(t) + Type(1e-8)), log(cots_pred[t] +
Type(1e-8)), sigma, true);

 // Likelihood contributions for coral observations
 Type sigma_coral = 0.1;
```

```
    nll -= dnorm(log(fast_dat(t) + Type(1e-8)), log(fast_pred[t] +
   Type(1e-8)), sigma_coral, true);
    nll -= dnorm(log(slow_dat(t) + Type(1e-8)), log(slow_pred[t] +
   Type(1e-8)), sigma_coral, true);
  }

  // Report predictions for external diagnostics.
  REPORT(cots_pred);
  REPORT(fast_pred);
  REPORT(slow_pred);

  return nll;
}
```

S7.1.3 Model Parameters


```
{
    "parameters": [
        {
            "parameter": "log_growth_rate",
            "value": 0.0,
            "description": "Logarithm of the intrinsic growth rate
   of COTS outbreaks (log(year^-1)).",
            "source": "initial estimate",
            "import_type": "PARAMETER",
            "priority": 1,
            "enhanced_semantic_description": "Crown-of-thorns
   starfish population exponential growth potential rate",
            "processed": true
        },
        {
            "parameter": "log_decline_rate",
            "value": 0.0,
            "description": "Logarithm of the outbreak decline rate (
   log(year^-1)).",
            "source": "initial estimate",
            "import_type": "PARAMETER",
            "priority": 1,
            "enhanced_semantic_description": "COTS population decay
   and mortality dynamics in marine ecosystems",
            "processed": true
        },
```

```
    {
        "parameter": "log_threshold",
        "value": 0.0,
        "description": "Logarithm of the threshold resource
level triggering outbreak (log(units)).",
        "source": "initial estimate",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Ecological tipping
point for triggering massive Crown-of-thorns starfish outbreak",
        "processed": true
    },
    {
        "parameter": "efficiency_fast",
        "value": 0.4,
        "description": "Efficiency factor for predation on fast-
growing coral communities (unitless). Lower value indicates
higher resilience.",
        "source": "expert opinion",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Predation impact
coefficient of starfish on fast-growing Acropora species",
        "processed": true
    },
    {
        "parameter": "efficiency_slow",
        "value": 0.6,
        "description": "Efficiency factor for predation on slow-
growing coral communities (unitless). Higher value reflects
increased vulnerability.",
        "source": "expert opinion",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Predation impact
coefficient of starfish on slow-growing Faviidae and Porites
species",
        "processed": true
    },
    {
        "parameter": "effect_sst",
        "value": 0.0,
```

```
        "description": "Effect of sea surface temperature on
outbreak progression (per \u00b0C).",
        "source": "expert opinion",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Sea surface
temperature's influence on marine predator population dynamics",
        "processed": true
    },
    {
        "parameter": "log_saturation",
        "value": 0.0,
        "description": "Logarithm of the saturation constant for
 resource limitation (log(units)).",
        "source": "initial estimate",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Resource limitation
and carrying capacity for marine predator populations",
        "processed": true
    },
    {
        "parameter": "log_outbreak_steepness",
        "value": 0.0,
        "description": "Logarithm of the outbreak steepness
coefficient controlling sensitivity of outbreak triggering to
resource limitation (log(unitless)).",
        "source": "model enhancement",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Determines how sharply
 outbreak growth is triggered as coral resource availability
exceeds the threshold",
        "processed": true
    },
    {
        "parameter": "temp_mod_eff_slow",
        "value": 0.0,
        "description": "Temperature modifier for predation
efficiency on slow-growing coral. Increasing temperatures may
amplify predation pressure due to reduced coral resilience.",
        "source": "model enhancement",
        "import_type": "PARAMETER",
```

```
            "priority": 3,
            "enhanced_semantic_description": "Modifies the
    effectiveness of starfish predation on slow-growing corals in
    response to sea surface temperature",
            "processed": true
        },
        {
            "parameter": "poly_coeff",
            "value": 0.0,
            "description": "Coefficient for quadratic adjustment to
    outbreak triggering when coral cover exceeds threshold. A
    positive value amplifies outbreak growth, enhancing boom dynamics
    .",
            "source": "model enhancement",
            "import_type": "PARAMETER",
            "priority": 3,
            "enhanced_semantic_description": "Higher-order response
    to excess coral resources driving rapid outbreak escalation",
            "processed": true
        }
    ]
}
```

S7.2 Claude 3.6 Sonnet Model

This model achieved an objective value of 0.6599.

S7.2.1 Model Intention

```
This model simulates the population dynamics of Crown of Thorns
    Starfish (COTS) and their interactions with coral communities on
    the Great Barrier Reef. The model captures:

1. Temperature-dependent COTS larval survival and recruitment
2. Density-dependent COTS population growth
3. Differential predation on fast-growing (Acropora) vs slow-growing
     (Faviidae/Porites) corals
4. Coral growth and recovery dynamics
5. Environmental thresholds that trigger outbreak events

Key mechanisms:
- COTS outbreaks are triggered by enhanced larval survival at
    optimal temperatures
- Population growth is limited by coral availability (resource
    limitation)
- Coral recovery is modeled with logistic growth, modified by COTS
    predation
- System includes feedback loops where coral depletion eventually
    limits COTS population growth

The model uses time series data of COTS abundance, coral cover (fast
    /slow-growing), sea surface temperature, and larval immigration
    rates to predict outbreak dynamics.
```

S7.2.2 Model Implementation

```
#include <TMB.hpp>
template<class Type>
Type objective_function<Type>::operator() ()
{
  // Data
  DATA_VECTOR(Year);                  // Time vector (years)
  DATA_VECTOR(sst_dat);              // Sea surface temperature (°C)
  DATA_VECTOR(cotsimm_dat);          // COTS larval immigration rate
     (individuals/m2/year)
```

```
DATA_VECTOR(cots_dat);              // Observed COTS density (
  individuals/m2)
DATA_VECTOR(fast_dat);              // Observed fast-growing coral
  cover (%)
DATA_VECTOR(slow_dat);              // Observed slow-growing coral
  cover (%)

// Parameters
PARAMETER(log_r_cots);              // COTS population growth rate
PARAMETER(log_K_cots);              // COTS carrying capacity
PARAMETER(log_temp_opt);            // Optimal temperature for COTS
   survival
PARAMETER(log_temp_range);          // Temperature tolerance range
PARAMETER(log_grazing_fast);        // Grazing rate on fast corals
PARAMETER(log_grazing_slow);        // Grazing rate on slow corals
PARAMETER(log_r_fast);              // Fast coral growth rate
PARAMETER(log_r_slow);              // Slow coral growth rate
PARAMETER(logit_coral_limit);       // Total coral cover limit
PARAMETER(log_obs_sd_cots);         // Observation error SD for
  COTS
PARAMETER(log_obs_sd_fast);         // Observation error SD for
  fast coral
PARAMETER(log_obs_sd_slow);         // Observation error SD for
  slow coral

// Transform parameters
Type r_cots = exp(log_r_cots);
Type K_cots = exp(log_K_cots);
Type temp_opt = exp(log_temp_opt);
Type temp_range = exp(log_temp_range);
Type grazing_fast = exp(log_grazing_fast);
Type grazing_slow = exp(log_grazing_slow);
Type r_fast = exp(log_r_fast);
Type r_slow = exp(log_r_slow);
Type coral_limit = invlogit(logit_coral_limit);
Type obs_sd_cots = exp(log_obs_sd_cots);
Type obs_sd_fast = exp(log_obs_sd_fast);
Type obs_sd_slow = exp(log_obs_sd_slow);

// Initialize predicted vectors
int n = Year.size();
vector<Type> cots_pred(n);
vector<Type> fast_pred(n);
```

```
vector<Type> slow_pred(n);

// Initialize negative log-likelihood
Type nll = 0.0;

// Small constant to prevent division by zero
Type eps = Type(1e-8);

// Set initial conditions
cots_pred(0) = cots_dat(0);
fast_pred(0) = fast_dat(0);
slow_pred(0) = slow_dat(0);

// Time series predictions
for(int t = 1; t < n; t++) {
  // 1. Temperature effect on COTS survival (Gaussian response)
  Type temp_effect = exp(-0.5 * pow((sst_dat(t-1) - temp_opt) /
 temp_range, 2));

  // 2. Resource limitation based on total coral cover
  Type total_coral = (fast_pred(t-1) + slow_pred(t-1)) / 100.0; //
   Convert to proportion
  Type resource_limit = total_coral / (total_coral + eps);

  // 3. COTS population dynamics with temperature-dependent
 recruitment
  Type recruitment = cotsimm_dat(t-1) * temp_effect;
  cots_pred(t) = cots_pred(t-1) * (1 + r_cots * resource_limit *
 (1 - cots_pred(t-1)/K_cots)) + recruitment;
  cots_pred(t) = CppAD::CondExpGt(cots_pred(t), Type(0), cots_pred
 (t), Type(0));

  // 4. Coral dynamics with COTS predation
  Type coral_space = (1 - (fast_pred(t-1) + slow_pred(t-1))/100.0/
 coral_limit);
  coral_space = CppAD::CondExpGt(coral_space, Type(0), coral_space
 , Type(0));

  // Fast-growing coral
  Type fast_growth = r_fast * fast_pred(t-1) * coral_space;
  Type fast_pred_loss = grazing_fast * cots_pred(t-1) * fast_pred(
 t-1);
  fast_pred(t) = fast_pred(t-1) + fast_growth - fast_pred_loss;
```

```
    fast_pred(t) = CppAD::CondExpGt(fast_pred(t), Type(0), fast_pred
   (t), Type(0));

    // Slow-growing coral
    Type slow_growth = r_slow * slow_pred(t-1) * coral_space;
    Type slow_pred_loss = grazing_slow * cots_pred(t-1) * slow_pred(
   t-1);
    slow_pred(t) = slow_pred(t-1) + slow_growth - slow_pred_loss;
    slow_pred(t) = CppAD::CondExpGt(slow_pred(t), Type(0), slow_pred
   (t), Type(0));
  }

  // Observation model using log-normal distribution
  for(int t = 0; t < n; t++) {
    // Add small constant to prevent log(0)
    nll -= dnorm(log(cots_dat(t) + eps), log(cots_pred(t) + eps),
   obs_sd_cots, true);
    nll -= dnorm(log(fast_dat(t) + eps), log(fast_pred(t) + eps),
   obs_sd_fast, true);
    nll -= dnorm(log(slow_dat(t) + eps), log(slow_pred(t) + eps),
   obs_sd_slow, true);
  }

  // Report predictions
  REPORT(cots_pred);
  REPORT(fast_pred);
  REPORT(slow_pred);

  return nll;
}
```

S7.2.3 Model Parameters


```
{
    "parameters": [
        {
            "parameter": "log_r_cots",
            "value": -0.223,
            "description": "Log of COTS population growth rate (year
   ^-1)",
            "source": "literature",
            "import_type": "PARAMETER",
```

```
        "priority": 1,
        "enhanced_semantic_description": "Crown-of-thorns
starfish population exponential growth potential",
        "citations": [
            "https://www.mdpi.com/1424-2818/9/1/1",
            "https://www.nature.com/articles/s41598-023-46749-x
",
            "https://royalsocietypublishing.org/doi/pdf/10.1098/
rspb.2020.1052"
        ],
        "processed": true
    },
    {
        "parameter": "log_K_cots",
        "value": 0.693,
        "description": "Log of COTS carrying capacity (
individuals/m2)",
        "source": "expert opinion",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Maximum sustainable
Crown-of-thorns starfish population density",
        "processed": true
    },
    {
        "parameter": "log_temp_opt",
        "value": 3.332,
        "description": "Log of optimal temperature for COTS
survival (\u00b0C)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 1,
        "enhanced_semantic_description": "Optimal sea
temperature for Crown-of-thorns starfish survival",
        "citations": [
            "https://link.springer.com/article/10.1007/s00227
-022-04027-w",
            "https://www.researchgate.net/publication/358986050
_Effects_of_elevated_temperature_on_the_performance_and_survival_of_pacific_crown
-of-thorns_starfish_Acanthaster_cf_solaris",
            "https://www.sciencedirect.com/science/article/abs/
pii/S0141113620305328"
        ],
```

```
        "processed": true,
        "found_value": 1.41,
        "found_min": 1.38,
        "found_max": 1.46
    },
    {
        "parameter": "log_temp_range",
        "value": 1.099,
        "description": "Log of temperature tolerance range (\
u00b0C)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Thermal tolerance
range for Crown-of-thorns starfish population",
        "citations": [
            "https://pubmed.ncbi.nlm.nih.gov/29281400/",
            "https://link.springer.com/article/10.1007/s00338
-013-1112-3",
            "https://link.springer.com/article/10.1007/s00227
-022-04027-w"
        ],
        "processed": true,
        "found_value": 1.39,
        "found_min": 1.26,
        "found_max": 1.49
    },
    {
        "parameter": "log_grazing_fast",
        "value": -2.303,
        "description": "Log of grazing rate on fast-growing
corals (m2/individual/year)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 1,
        "enhanced_semantic_description": "Predation impact on
fast-growing coral species by starfish",
        "citations": [
            "https://www.sciencedirect.com/science/article/pii/
S0048969724028389",
            "https://pubmed.ncbi.nlm.nih.gov/38663591/",
            "https://www.aims.gov.au/research-topics/marine-life
/crown-thorns-starfish"
```

```
        ],
        "processed": true
    },
    {
        "parameter": "log_grazing_slow",
        "value": -2.996,
        "description": "Log of grazing rate on slow-growing
corals (m2/individual/year)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 1,
        "enhanced_semantic_description": "Predation impact on
slow-growing coral species by starfish",
        "citations": [
            "https://www.sciencedirect.com/science/article/abs/
pii/S0022519384801381",
            "https://onlinelibrary.wiley.com/doi/10.1111/j
.1461-0248.2004.00593.x",
            "https://www.int-res.com/articles/theme/m512p167.pdf
"
        ],
        "processed": true
    },
    {
        "parameter": "log_r_fast",
        "value": -0.916,
        "description": "Log of fast coral growth rate (year^-1)
",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Rapid coral species
recovery and regeneration potential",
        "citations": [
            "https://www.nature.com/articles/s41598
-025-93531-2",
            "https://www.sciencedirect.com/science/article/pii/
S0960982224001519",
            "https://news.mongabay.com/short-article/corals-
recover-faster-on-artificial-structures-than-on-natural-reefs-
study-finds/"
        ],
        "processed": true
```

```
    },
    {
        "parameter": "log_r_slow",
        "value": -1.609,
        "description": "Log of slow coral growth rate (year^-1)
",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Slow coral species
recovery and regeneration potential",
        "citations": [
            "https://esajournals.onlinelibrary.wiley.com/doi
/10.1002/ecs2.4915",
            "https://besjournals.onlinelibrary.wiley.com/doi/
full/10.1111/1365-2664.13141",
            "https://www.sciencedirect.com/science/article/pii/
S0141113618305002"
        ],
        "processed": true
    },
    {
        "parameter": "logit_coral_limit",
        "value": 1.386,
        "description": "Logit of maximum total coral cover as
proportion",
        "source": "expert opinion",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Maximum ecological
carrying capacity for total coral coverage",
        "processed": true
    },
    {
        "parameter": "log_obs_sd_cots",
        "value": -1.609,
        "description": "Log of observation error SD for COTS
abundance",
        "source": "initial estimate",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Measurement
uncertainty in Crown-of-thorns starfish population estimates",
```

```
            "processed": true
        },
        {
            "parameter": "log_obs_sd_fast",
            "value": -1.609,
            "description": "Log of observation error SD for fast
    coral cover",
            "source": "initial estimate",
            "import_type": "PARAMETER",
            "priority": 3,
            "enhanced_semantic_description": "Measurement
    uncertainty in fast-growing coral cover estimates",
            "processed": true
        },
        {
            "parameter": "log_obs_sd_slow",
            "value": -1.609,
            "description": "Log of observation error SD for slow
    coral cover",
            "source": "initial estimate",
            "import_type": "PARAMETER",
            "priority": 3,
            "enhanced_semantic_description": "Measurement
    uncertainty in slow-growing coral cover estimates",
            "processed": true
        }
    ]
}
```

S7.3 Claude 3.7 Sonnet Model

This model achieved an objective value of 0.5204.

S7.3.1 Model Intention

```
This model simulates the population dynamics of Crown of Thorns
    Starfish (COTS) and their impact on coral communities in the
    Great Barrier Reef. The model captures the characteristic boom-
    bust cycles of COTS outbreaks while accounting for their
    selective predation on different coral functional groups.

Key ecological processes included:
1. COTS population dynamics with density-dependent mortality and
    recruitment influenced by sea surface temperature
2. Differential predation by COTS on fast-growing (Acropora spp.)
    and slow-growing (Faviidae and Porites spp.) corals
3. Coral growth and recovery dynamics with competition between coral
     functional groups
4. Environmental forcing through sea surface temperature effects on
    COTS recruitment and coral mortality
5. Episodic COTS larval immigration events that can trigger
    outbreaks

The model uses a system of coupled differential equations
    discretized to an annual time step, with parameters estimated
    from time series data. Functional responses include saturating
    predation rates, temperature-dependent recruitment, and density-
    dependent mortality. The model aims to predict the timing,
    magnitude, and duration of COTS outbreaks and their cascading
    effects on coral community composition.
```

S7.3.2 Model Implementation

```
#include <TMB.hpp>

template<class Type>
Type objective_function<Type>::operator() ()
{
  // DATA
  DATA_VECTOR(Year);                    // Years of observation
  DATA_VECTOR(cots_dat);                // Observed COTS abundance (
    individuals/m^2)
```

```
DATA_VECTOR(fast_dat);              // Observed fast-growing coral
   cover (%)
DATA_VECTOR(slow_dat);              // Observed slow-growing coral
   cover (%)
DATA_VECTOR(sst_dat);               // Sea surface temperature (°C)
DATA_VECTOR(cotsimm_dat);           // COTS larval immigration
  rate (individuals/m^2/year)

// PARAMETERS
PARAMETER(r_cots);                  // Intrinsic growth rate of
  COTS population (year^-1)
PARAMETER(K_cots);                  // Carrying capacity of COTS
  population (individuals/m^2)
PARAMETER(m_cots);                  // Natural mortality rate of
  COTS (year^-1)
PARAMETER(r_fast);                  // Intrinsic growth rate of
  fast-growing coral (year^-1)
PARAMETER(K_fast);                  // Maximum cover of fast-
  growing coral (%)
PARAMETER(r_slow);                  // Intrinsic growth rate of
  slow-growing coral (year^-1)
PARAMETER(K_slow);                  // Maximum cover of slow-
  growing coral (%)
PARAMETER(a_fast);                  // Attack rate of COTS on fast
  -growing coral (m^2/individual/year)
PARAMETER(a_slow);                  // Attack rate of COTS on slow
  -growing coral (m^2/individual/year)
PARAMETER(h_fast);                  // Handling time for COTS
  feeding on fast-growing coral (% cover)
PARAMETER(h_slow);                  // Handling time for COTS
  feeding on slow-growing coral (% cover)
PARAMETER(temp_opt);                // Optimal temperature for
  COTS recruitment (°C)
PARAMETER(temp_width);              // Temperature range width for
   COTS recruitment (°C)
PARAMETER(imm_effect);              // Effect of larval
  immigration on COTS recruitment (dimensionless)
PARAMETER(competition);             // Competition coefficient
  between coral types (dimensionless)
PARAMETER(bleach_threshold);        // Temperature threshold for
  coral bleaching (°C)
PARAMETER(bleach_mortality_fast);   // Mortality rate of fast-
  growing coral during bleaching (year^-1)
```

```
PARAMETER(bleach_mortality_slow);   // Mortality rate of slow-
  growing coral during bleaching (year^-1)
PARAMETER(sigma_cots);              // Observation error standard
  deviation for COTS abundance (log scale)
PARAMETER(sigma_fast);              // Observation error standard
  deviation for fast-growing coral cover (log scale)
PARAMETER(sigma_slow);              // Observation error standard
  deviation for slow-growing coral cover (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_years = Year.size();

// Vectors to store model predictions
vector<Type> cots_pred(n_years);
vector<Type> fast_pred(n_years);
vector<Type> slow_pred(n_years);

// Initialize with first year's data
cots_pred(0) = cots_dat(0);
fast_pred(0) = fast_dat(0);
slow_pred(0) = slow_dat(0);

// Minimum standard deviations to prevent numerical issues
Type min_sigma = Type(0.01);
Type sigma_cots_adj = sigma_cots + min_sigma;
Type sigma_fast_adj = sigma_fast + min_sigma;
Type sigma_slow_adj = sigma_slow + min_sigma;

// Time series simulation
for (int t = 1; t < n_years; t++) {
  // Previous time step values
  Type cots_t0 = cots_pred(t-1);
  Type fast_t0 = fast_pred(t-1);
  Type slow_t0 = slow_pred(t-1);
  Type sst = sst_dat(t-1);
  Type cotsimm = cotsimm_dat(t-1);
```

```
// 1. Temperature effect on COTS recruitment
// Gaussian response curve for temperature effect on COTS
recruitment
Type temp_effect = exp(-0.5 * pow((sst - temp_opt) / temp_width,
 2));

// 2. COTS functional response (Type II) for predation on corals
// Holling Type II functional response for COTS predation on
fast-growing coral
Type pred_fast = (a_fast * fast_t0 * cots_t0) / (1.0 + a_fast *
h_fast * fast_t0 + a_slow * h_slow * slow_t0 + eps);

// Holling Type II functional response for COTS predation on
slow-growing coral
Type pred_slow = (a_slow * slow_t0 * cots_t0) / (1.0 + a_fast *
h_fast * fast_t0 + a_slow * h_slow * slow_t0 + eps);

// 3. Bleaching effect on corals
// Smooth transition function for bleaching effect
Type bleach_effect = 1.0 / (1.0 + exp(-2.0 * (sst -
bleach_threshold)));

// 4. COTS population dynamics
// COTS population growth with density dependence, temperature
effect on recruitment, and immigration
Type cots_growth = r_cots * cots_t0 * (1.0 - cots_t0 / K_cots) *
 temp_effect;

// Immigration effect with smooth transition
Type imm_term = imm_effect * cotsimm / (1.0 + cotsimm + eps);

// Food limitation effect (COTS mortality increases when coral
cover is low)
Type food_limitation = m_cots * (1.0 + 1.0 / (fast_t0 + slow_t0
+ eps));

// Update COTS abundance
cots_pred(t) = cots_t0 + cots_growth - food_limitation * cots_t0
 + imm_term;
cots_pred(t) = cots_pred(t) < eps ? eps : cots_pred(t); //
Ensure positive values

// 5. Coral dynamics
```

```
  // Fast-growing coral dynamics with logistic growth, competition
  , predation, and bleaching
  Type fast_growth = r_fast * fast_t0 * (1.0 - (fast_t0 +
  competition * slow_t0) / K_fast);
  Type fast_bleaching = bleach_mortality_fast * bleach_effect *
  fast_t0;

  // Update fast-growing coral cover
  fast_pred(t) = fast_t0 + fast_growth - pred_fast -
  fast_bleaching;
  fast_pred(t) = fast_pred(t) < eps ? eps : fast_pred(t); //
  Ensure positive values

  // Slow-growing coral dynamics with logistic growth, competition
  , predation, and bleaching
  Type slow_growth = r_slow * slow_t0 * (1.0 - (slow_t0 +
  competition * fast_t0) / K_slow);
  Type slow_bleaching = bleach_mortality_slow * bleach_effect *
  slow_t0;

  // Update slow-growing coral cover
  slow_pred(t) = slow_t0 + slow_growth - pred_slow -
  slow_bleaching;
  slow_pred(t) = slow_pred(t) < eps ? eps : slow_pred(t); //
  Ensure positive values
}

// Calculate negative log-likelihood using lognormal distribution
for (int t = 0; t < n_years; t++) {
  // Add small constant to data and predictions to handle zeros
  Type cots_obs = cots_dat(t) + eps;
  Type cots_mod = cots_pred(t) + eps;
  Type fast_obs = fast_dat(t) + eps;
  Type fast_mod = fast_pred(t) + eps;
  Type slow_obs = slow_dat(t) + eps;
  Type slow_mod = slow_pred(t) + eps;

  // Log-normal likelihood for COTS abundance
  nll -= dnorm(log(cots_obs), log(cots_mod), sigma_cots_adj, true)
  ;

  // Log-normal likelihood for fast-growing coral cover
```

```
    nll -= dnorm(log(fast_obs), log(fast_mod), sigma_fast_adj, true)
    ;

    // Log-normal likelihood for slow-growing coral cover
    nll -= dnorm(log(slow_obs), log(slow_mod), sigma_slow_adj, true)
    ;
  }

  // Add penalties for biologically implausible parameter values
  // Smooth penalty for growth rates to be positive
  nll += Type(10.0) / (Type(1.0) + exp(Type(5.0) * r_cots));
  nll += Type(10.0) / (Type(1.0) + exp(Type(5.0) * r_fast));
  nll += Type(10.0) / (Type(1.0) + exp(Type(5.0) * r_slow));

  // Report predictions
  REPORT(cots_pred);
  REPORT(fast_pred);
  REPORT(slow_pred);

  return nll;
}
```

S7.3.3 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 exponential growth potential",
            "citations": [
                "https://www.mdpi.com/1424-2818/9/1/1",
                "https://www.nature.com/articles/s41598-023-46749-x
   ",
                "https://royalsocietypublishing.org/doi/pdf/10.1098/
   rspb.2020.1052"
```

```
        ],
        "processed": true
    },
    {
        "parameter": "K_cots",
        "value": 2.5,
        "description": "Carrying capacity of COTS population (
individuals/m^2)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Maximum sustainable
population density for crown-of-thorns starfish",
        "citations": [
            "https://www.nature.com/articles/s41598-023-46749-x
",
            "https://www.sciencedirect.com/science/article/abs/
pii/S0964569112002669",
            "https://researchonline.jcu.edu.au/24119/2/02whole.
pdf"
        ],
        "processed": true,
        "found_value": 0.002075,
        "found_min": 0.00015,
        "found_max": 0.004
    },
    {
        "parameter": "m_cots",
        "value": 0.4,
        "description": "Natural mortality rate of COTS (year^-1)
",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Natural death 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_popula
",
```

```
            "https://en.wikipedia.org/wiki/Crown-of-
thorns_starfish"
        ],
        "processed": true,
        "found_value": 6.0,
        "found_min": 2.49,
        "found_max": 9.49
    },
    {
        "parameter": "r_fast",
        "value": 0.3,
        "description": "Intrinsic growth rate of fast-growing
coral (year^-1)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 1,
        "enhanced_semantic_description": "Rapid coral growth
rate for fast-colonizing species",
        "citations": [
            "https://besjournals.onlinelibrary.wiley.com/doi/
full/10.1111/2041-210X.13388",
            "https://coralcavern.com/how-fast-does-coral-grow-in
-colonies/",
            "https://link.springer.com/article/10.1007/s00227
-024-04511-5"
        ],
        "processed": true,
        "found_value": 0.72,
        "found_min": 0.06,
        "found_max": 2.0
    },
    {
        "parameter": "K_fast",
        "value": 50.0,
        "description": "Maximum cover of fast-growing coral (%)
",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Maximum percentage
coverage for fast-growing coral species",
        "citations": [
```

```
            "https://www.researchgate.net/figure/Percentage-
coverage-of-coral-families_tbl1_329554775",
            "https://www.epd.gov.hk/eia/register/report/
eiareport/eia_1482008/EIA/html/Text/S11_Marine%20Ecology.htm",
            "https://link.springer.com/article/10.1007/s00227
-024-04511-5"
        ],
        "processed": true,
        "found_value": 41.8,
        "found_min": 10.3,
        "found_max": 75.0
    },
    {
        "parameter": "r_slow",
        "value": 0.1,
        "description": "Intrinsic growth rate of slow-growing
coral (year^-1)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 1,
        "enhanced_semantic_description": "Gradual coral growth
rate for slow-colonizing species",
        "citations": [
            "https://www.frontiersin.org/journals/marine-science
/articles/10.3389/fmars.2020.00483/full",
            "https://besjournals.onlinelibrary.wiley.com/doi/
full/10.1111/2041-210X.13388",
            "https://www.int-res.com/articles/meps/126/m126p145.
pdf"
        ],
        "processed": true
    },
    {
        "parameter": "K_slow",
        "value": 30.0,
        "description": "Maximum cover of slow-growing coral (%)
",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Maximum percentage
coverage for slow-growing coral species",
        "citations": [
```

```
            "https://www.researchgate.net/figure/Percentage-
coverage-of-coral-families_tbl1_329554775",
            "https://www.sciencedirect.com/science/article/abs/
pii/S0006320799000671",
            "https://link.springer.com/article/10.1007/s00338
-024-02602-9"
        ],
        "processed": true,
        "found_value": 15.0,
        "found_min": 0.0,
        "found_max": 40.0
    },
    {
        "parameter": "a_fast",
        "value": 0.2,
        "description": "Attack rate of COTS on fast-growing
coral (m^2/individual/year)",
        "source": "expert opinion",
        "import_type": "PARAMETER",
        "priority": 1,
        "enhanced_semantic_description": "Predation intensity of
 starfish on fast-growing coral",
        "processed": true
    },
    {
        "parameter": "a_slow",
        "value": 0.05,
        "description": "Attack rate of COTS on slow-growing
coral (m^2/individual/year)",
        "source": "expert opinion",
        "import_type": "PARAMETER",
        "priority": 1,
        "enhanced_semantic_description": "Predation intensity of
 starfish on slow-growing coral",
        "processed": true
    },
    {
        "parameter": "h_fast",
        "value": 10.0,
        "description": "Handling time for COTS feeding on fast-
growing coral (% cover)",
        "source": "initial estimate",
        "import_type": "PARAMETER",
```

```
        "priority": 3,
        "enhanced_semantic_description": "Feeding consumption
time for fast-growing coral species",
        "processed": true
    },
    {
        "parameter": "h_slow",
        "value": 15.0,
        "description": "Handling time for COTS feeding on slow-
growing coral (% cover)",
        "source": "initial estimate",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Feeding consumption
time for slow-growing coral species",
        "processed": true
    },
    {
        "parameter": "temp_opt",
        "value": 28.0,
        "description": "Optimal temperature for COTS recruitment
 (\u00b0C)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Ideal temperature
range for crown-of-thorns starfish recruitment",
        "citations": [
            "https://www.researchgate.net/publication/358986050
_Effects_of_elevated_temperature_on_the_performance_and_survival_of_pacific_crown
-of-thorns_starfish_Acanthaster_cf_solaris",
            "https://www.jcu.edu.au/news/releases/2022/december/
coral-eating-starfish-another-victim-of-climate-change",
            "https://pmc.ncbi.nlm.nih.gov/articles/PMC4325318/"
        ],
        "processed": true,
        "found_value": 28.0,
        "found_min": 26.0,
        "found_max": 30.0
    },
    {
        "parameter": "temp_width",
        "value": 2.0,
```

```
        "description": "Temperature range width for COTS
recruitment (\u00b0C)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Temperature tolerance
range for starfish population dynamics",
        "citations": [
            "https://link.springer.com/article/10.1007/s00227
-022-04027-w",
            "https://www.sciencedirect.com/science/article/abs/
pii/S0022098101002982",
            "https://www.researchgate.net/publication/358986050
_Effects_of_elevated_temperature_on_the_performance_and_survival_of_pacific_crown
-of-thorns_starfish_Acanthaster_cf_solaris"
        ],
        "processed": true,
        "found_value": 28.0,
        "found_min": 26.0,
        "found_max": 30.0
    },
    {
        "parameter": "imm_effect",
        "value": 0.8,
        "description": "Effect of larval immigration on COTS
recruitment (dimensionless)",
        "source": "expert opinion",
        "import_type": "PARAMETER",
        "priority": 1,
        "enhanced_semantic_description": "Larval immigration
impact on starfish population recruitment",
        "processed": true
    },
    {
        "parameter": "competition",
        "value": 0.3,
        "description": "Competition coefficient between coral
types (dimensionless)",
        "source": "initial estimate",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Inter-species
competitive interaction between different coral types",
```

```
        "processed": true
    },
    {
        "parameter": "bleach_threshold",
        "value": 30.0,
        "description": "Temperature threshold for coral
bleaching (\u00b0C)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Critical temperature
triggering widespread coral bleaching event",
        "citations": [
            "https://www.downtoearth.org.in/climate-change/
scientists-may-take-at-least-a-year-to-grasp-full-scale-of-worlds
-largest-coral-bleaching-event",
            "https://www.coris.noaa.gov/activities/
reef_managers_guide/reef_managers_guide_ch4.pdf",
            "https://www.nature.com/articles/s43247-025-02195-3"
        ],
        "processed": true,
        "found_value": 1.5,
        "found_min": 1.0,
        "found_max": 2.0
    },
    {
        "parameter": "bleach_mortality_fast",
        "value": 0.8,
        "description": "Mortality rate of fast-growing coral
during bleaching (year^-1)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Mortality rate of fast
-growing corals during bleaching stress",
        "citations": [
            "https://journals.plos.org/plosone/article?id
=10.1371/journal.pone.0322636",
            "https://pmc.ncbi.nlm.nih.gov/articles/PMC6899343/",
            "https://www.sciencedirect.com/science/article/abs/
pii/S0025326X17305829"
        ],
        "processed": true,
```

```
        "found_value": 0.585,
        "found_min": 0.114,
        "found_max": 0.8
    },
    {
        "parameter": "bleach_mortality_slow",
        "value": 0.3,
        "description": "Mortality rate of slow-growing coral
during bleaching (year^-1)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Mortality rate of slow
-growing corals during bleaching stress",
        "citations": [
            "https://www.sciencedirect.com/science/article/abs/
pii/S0048969722083310",
            "https://pmc.ncbi.nlm.nih.gov/articles/PMC6899343/",
            "https://onlinelibrary.wiley.com/doi/10.1111/eva
.13500"
        ],
        "processed": true,
        "found_value": 0.185,
        "found_min": 0.04,
        "found_max": 0.415
    },
    {
        "parameter": "sigma_cots",
        "value": 0.2,
        "description": "Observation error standard deviation for
 COTS abundance (log scale)",
        "source": "initial estimate",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Measurement
uncertainty for crown-of-thorns starfish population abundance",
        "processed": true
    },
    {
        "parameter": "sigma_fast",
        "value": 0.3,
        "description": "Observation error standard deviation for
 fast-growing coral cover (log scale)",
```

```
            "source": "initial estimate",
            "import_type": "PARAMETER",
            "priority": 3,
            "enhanced_semantic_description": "Measurement
    uncertainty for fast-growing coral species coverage",
            "processed": true
        },
        {
            "parameter": "sigma_slow",
            "value": 0.3,
            "description": "Observation error standard deviation for
     slow-growing coral cover (log scale)",
            "source": "initial estimate",
            "import_type": "PARAMETER",
            "priority": 3,
            "enhanced_semantic_description": "Measurement
    uncertainty for slow-growing coral species coverage",
            "processed": true
        }
    ]
}
```

S7.4 o4 mini Model

This model achieved an objective value of 0.3486.

S7.4.1 Model Intention

S7.4.2 Model Implementation

```
#include <TMB.hpp> // TMB header: provides templated functions and
    macros for the model

// Model Overview:
// 1. COTS Dynamics: Logistic growth modified by a smooth outbreak-
    triggered decline.
// 2. Fast-growing Coral (Acropora spp.) Dynamics: Recovery towards
    full cover minus losses due to COTS predation.
// 3. Slow-growing Coral (Faviidae spp. and Porites spp.) Dynamics:
    Similar recovery with slower dynamics and losses by COTS.
// Each parameter is commented with its units, origin, and role in
    the ecological processes.

template<class Type>
Type objective_function<Type>::operator() () {
    // --- DATA INPUTS ---
    // time: vector of years (as provided in the first column of the
     CSV)
    DATA_VECTOR(time);                    // (years)
    DATA_VECTOR(cots_dat);                // Observed COTS abundance
    (individuals/m2)
    DATA_VECTOR(fast_dat);                // Observed fast-growing
    coral cover (%) for Acropora spp.
    DATA_VECTOR(slow_dat);                // Observed slow-growing
    coral cover (%) for Faviidae spp. & Porites spp.

    // --- PARAMETERS ---
    // growth_rate_cots: Intrinsic growth rate of COTS (year^-1)
    // decay_rate_cots: Decline rate of COTS post-outbreak (year^-1)
    // coral_predation_efficiency: Efficiency of COTS predation on
    coral communities (per individual/m2)
    // carrying_capacity: Ecosystem carrying capacity for COTS (
    individuals/m2)
```

```
// observed_sd: Standard deviation for lognormal observation
errors
PARAMETER(growth_rate_cots);         // (year^-1), literature/
expert opinion
PARAMETER(decay_rate_cots);          // (year^-1), literature
PARAMETER(coral_predation_efficiency); // (m2/individual),
expert opinion
PARAMETER(carrying_capacity);        // (individuals/m2),
literature
PARAMETER(observed_sd);              // (log-scale units),
initial estimate
PARAMETER(outbreak_sharpness);       // (unitless), governs the
steepness of the outbreak trigger function
PARAMETER(handling_time);             // (time units), handling
time for saturating predation response (Holling Type II)
PARAMETER(outbreak_threshold);       // (unitless), fraction of
carrying capacity for outbreak trigger
PARAMETER(outbreak_shape);           // (unitless), exponent
controlling non-linear outbreak trigger sensitivity - values >1
increase outbreak threshold sensitivity.
PARAMETER(outbreak_hysteresis);      // (unitless), captures
hysteresis in outbreak dynamics to model delayed decline post-
outbreak.
PARAMETER(outbreak_nonlinearity);    // (unitless), additional
non-linear amplification factor in the outbreak trigger function.
PARAMETER(outbreak_decline_exponent);       // (unitless),
exponent for non-linear outbreak decline dynamics. Values > 1
intensify the decline during outbreak.
PARAMETER(resource_limitation_strength);       // (unitless),
scaling factor representing effect of coral availability on COTS
growth
PARAMETER(environmental_modifier);
PARAMETER(predation_scaler);
PARAMETER(coral_recovery_modifier);
PARAMETER(coral_recovery_inhibition);
PARAMETER(coral_recovery_environmental_modifier);
PARAMETER(predation_efficiency_exponent);
PARAMETER(extreme_outbreak_modifier);
PARAMETER(coral_saturation_coefficient);

// --- NUMERICAL STABILITY ---
Type eps = Type(1e-8); // small constant to avoid division by
zero
```

```
int n = cots_dat.size();
// Vectors to hold predictions (suffix _pred corresponds to
observation names)
vector<Type> cots_pred(n);
vector<Type> fast_pred(n);
vector<Type> slow_pred(n);

// --- INITIAL CONDITIONS ---
cots_pred[0] = cots_dat[0];  // Use first observation as initial
 state
fast_pred[0] = fast_dat[0];
slow_pred[0] = slow_dat[0];

// --- MODEL EQUATIONS (loop over time steps; t uses previous
state only) ---
for(int t = 1; t < n; t++){
    // Equation 1: COTS Dynamics
    // Incorporate resource limitation by scaling growth with
available coral cover (sum of fast and slow predictions).
    Type coral_availability = (fast_pred[t-1] + slow_pred[t-1])
/ Type(200);
    // Introduce a saturating function with a quadratic term for
 coral availability to capture diminishing returns.
    Type resource_factor = (coral_availability +
coral_saturation_coefficient * pow(coral_availability, 2))
                           / (Type(0.5) + coral_availability +
coral_saturation_coefficient * pow(coral_availability, 2));
    Type growth = growth_rate_cots * cots_pred[t-1] * (1 -
cots_pred[t-1] / (carrying_capacity + eps)) * (Type(1) +
resource_limitation_strength * (resource_factor - Type(0.5)));
    Type effective_sharpness = outbreak_sharpness *
environmental_modifier * (Type(1) + extreme_outbreak_modifier * (
environmental_modifier - Type(1)));
    Type raw_trigger = 1 / (Type(1) + exp(- effective_sharpness
* ( pow(cots_pred[t-1], outbreak_shape) + outbreak_nonlinearity *
 pow(cots_pred[t-1], 2) - pow(outbreak_threshold *
carrying_capacity, outbreak_shape) )));
    Type outbreak_trigger = raw_trigger + outbreak_hysteresis *
raw_trigger * (Type(1) - raw_trigger);
    Type decline = decay_rate_cots * pow(cots_pred[t-1],
outbreak_decline_exponent) * outbreak_trigger;
```

```
    cots_pred[t] = cots_pred[t-1] + growth - decline; // Updated
 COTS population

    // Equation 2: Fast-growing Coral Dynamics (Acropora spp.)
    //  - Recovery: Proportional to the gap to maximum cover (
assumed 100%) modified by inhibition from high COTS levels
    //  - Decline: Losses due to predation by COTS with
saturation at low coral cover
    fast_pred[t] = fast_pred[t-1] + Type(0.1) *
coral_recovery_modifier * coral_recovery_environmental_modifier *
 (Type(100) - fast_pred[t-1]) * (Type(1) -
coral_recovery_inhibition * cots_pred[t-1] / (carrying_capacity +
 eps))
                 - (cots_pred[t-1] * coral_predation_efficiency
* fast_pred[t-1] *
                   pow((fast_pred[t-1] / (fast_pred[t-1] +
predation_scaler)), predation_efficiency_exponent)) / (Type(1) +
handling_time * fast_pred[t-1]);

    // Equation 3: Slow-growing Coral Dynamics (Faviidae/Porites
 spp.)
    //  - Recovery: Slower than fast-growing coral, modified by
inhibition from high COTS levels
    //  - Decline: Affected by COTS predation with saturation at
 low coral cover
    slow_pred[t] = slow_pred[t-1] + Type(0.05) *
coral_recovery_environmental_modifier * (Type(100) - slow_pred[t
-1]) * (Type(1) - coral_recovery_inhibition * cots_pred[t-1] / (
carrying_capacity + eps))
                 - (cots_pred[t-1] * coral_predation_efficiency
* slow_pred[t-1] *
                   pow((slow_pred[t-1] / (slow_pred[t-1] +
predation_scaler)), predation_efficiency_exponent)) / (Type(1) +
handling_time * slow_pred[t-1]);
}

// --- LIKELIHOOD CALCULATION ---
// Use lognormal distributions (log-transformed data) for
strictly positive observations.
// A fixed small standard deviation is enforced via observed_sd.
Type jnll = 0.0;
for(int t = 0; t < n; t++){
```

```
        jnll -= dnorm(log(cots_dat[t] + eps), log(cots_pred[t] + eps
    ), observed_sd, true);
        jnll -= dnorm(log(fast_dat[t] + eps), log(fast_pred[t] + eps
    ), observed_sd, true);
        jnll -= dnorm(log(slow_dat[t] + eps), log(slow_pred[t] + eps
    ), observed_sd, true);
    }

    // --- REPORTING ---
    // Report all predicted variables with the _pred suffix as
    required.
    REPORT(cots_pred);  // Predicted COTS
    REPORT(fast_pred);  // Predicted Fast-growing Coral Cover
    REPORT(slow_pred);  // Predicted Slow-growing Coral Cover

    return jnll;
}
```

S7.4.3 Model Parameters


```
{
    "parameters": [
        {
            "parameter": "growth_rate_cots",
            "value": 0.5,
            "description": "Intrinsic outbreak growth rate of COTS (
    year^-1). Literature/expert estimate.",
            "source": "literature",
            "import_type": "PARAMETER",
            "priority": 1,
            "enhanced_semantic_description": "Crown-of-thorns
    starfish population exponential growth dynamics",
            "citations": [
                "https://www.mdpi.com/1424-2818/9/1/1",
                "https://www.nature.com/articles/s41598-023-46749-x
    ",
                "https://royalsocietypublishing.org/doi/pdf/10.1098/
    rspb.2020.1052"
            ],
            "processed": true
        },
        {
```

```
        "parameter": "decay_rate_cots",
        "value": 0.3,
        "description": "Decay rate of COTS post-outbreak (year
^-1). Influences the bust phase.",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 1,
        "enhanced_semantic_description": "Population collapse
mechanism in marine ecosystem outbreak cycles",
        "citations": [
            "https://www.sciencedirect.com/science/article/abs/
pii/S0169534707003552",
            "https://royalsocietypublishing.org/doi/10.1098/rspb
.2017.2841",
            "https://esajournals.onlinelibrary.wiley.com/doi
/10.1002/ecs2.4580"
        ],
        "processed": true
    },
    {
        "parameter": "coral_predation_efficiency",
        "value": 0.05,
        "description": "Predation efficiency of COTS on coral
species (m2/individual). Based on expert opinion.",
        "source": "expert opinion",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Quantitative measure
of starfish destructive impact on coral reefs",
        "processed": true
    },
    {
        "parameter": "carrying_capacity",
        "value": 50,
        "description": "Carrying capacity for COTS (individuals/
m2). Determines saturation level in the ecosystem.",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 1,
        "enhanced_semantic_description": "Maximum sustainable
population density for marine predator species",
        "citations": [
```

"https://iopscience.iop.org/article
/10.1088/1742-6596/1280/2/022036/pdf",
"https://www.sciencedirect.com/science/article/abs/
pii/S0303264718304532",
"https://www.researchgate.net/publication/238373500
_Maximum_sustainable_yield_and_species_extinction_in_ecosystems"
],
"processed": true
},
{
"parameter": "observed_sd",
"value": 0.1,
"description": "Standard deviation for lognormal
observation error (log-scale). Fixed minimum to ensure numerical
stability.",
"source": "initial estimate",
"import_type": "PARAMETER",
"priority": 1,
"enhanced_semantic_description": "Statistical
uncertainty and variability in ecological population measurements
",
"processed": true
},
{
"parameter": "outbreak_sharpness",
"value": 100,
"description": "Controls outbreak trigger steepness;
higher values yield a more threshold-like response capturing
outbreak initiation dynamics.",
"source": "model refinement based on ecological feedback
",
"import_type": "PARAMETER",
"priority": 2,
"enhanced_semantic_description": "Sensitivity of
outbreak initiation relative to predator density",
"processed": true
},
{
"parameter": "handling_time",
"value": 0.01,
"description": "Handling time for coral predation:
accounts for the saturating functional response in COTS predation
on coral. A lower value implies quicker handling, leading to

```
less saturation.",
        "source": "model refinement based on ecological feedback
",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Non-linear saturation
in predation dynamics due to handling time",
        "processed": true
    },
    {
        "parameter": "environmental_modifier",
        "value": 1.0,
        "description": "Modifier for outbreak trigger steepness
representing environmental influences (e.g. temperature anomalies
) on outbreak timing.",
        "source": "model refinement with ecological feedback",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Multiplicative factor
modulating the outbreak trigger based on environmental conditions
",
        "processed": true
    },
    {
        "parameter": "predation_scaler",
        "value": 0.5,
        "description": "Scales coral predation efficiency to
reflect reduced effectiveness at low coral cover, representing
diminishing returns in prey detection or availability.",
        "source": "ecological model refinement",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Non-linear modulation
of coral predation efficiency based on coral cover",
        "processed": true
    },
    {
        "parameter": "coral_recovery_modifier",
        "value": 1.0,
        "description": "Modifier scaling coral recovery rates
based on favorable environmental conditions (e.g., water quality,
 temperature).",
```

```
        "source": "model refinement based on ecological feedback
",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Environmental
influence on coral recovery dynamics, capturing variable
ecosystem resilience",
        "processed": true
    },
    {
        "parameter": "coral_recovery_inhibition",
        "value": 0.2,
        "description": "Inhibition factor on coral recovery due
to sustained high COTS pressure. This factor reduces the recovery
 rate of corals when high COTS numbers persist, capturing delayed
 ecosystem resilience.",
        "source": "model refinement based on ecological feedback
",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Modulates coral
recovery in response to prolonged high predation pressure",
        "processed": true
    },
    {
        "parameter": "coral_recovery_environmental_modifier",
        "value": 1.0,
        "description": "Scaling factor that modulates coral
recovery rates based on environmental conditions (e.g., water
temperature, nutrients). It captures how favorable or unfavorable
 conditions affect coral resilience.",
        "source": "model refinement based on ecological feedback
",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Dynamic adjustment of
coral recovery rates in response to environmental variability",
        "processed": true
    },
    {
        "parameter": "outbreak_threshold",
        "value": 0.5,
```

```
        "description": "Threshold fraction of carrying capacity
at which an outbreak is triggered. This parameter allows tuning
the outbreak onset in response to variable ecological conditions
.",
        "source": "model refinement with ecological feedback",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Flexible threshold for
 triggering COTS outbreaks based on relative carrying capacity",
        "processed": true
    },
    {
        "parameter": "resource_limitation_strength",
        "value": 1.0,
        "description": "Scaling factor that modulates the effect
 of coral availability on COTS growth through a saturating
functional response, capturing diminishing returns as coral cover
 increases.",
        "source": "model refinement based on ecological feedback
",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Modulates COTS growth
via a Michaelis\u2013Menten type resource limitation effect",
        "processed": true
    },
    {
        "parameter": "outbreak_decline_exponent",
        "value": 1.0,
        "description": "Exponent for non-linear outbreak decline
. Values > 1 intensify the population collapse during outbreak
phases, capturing stronger density-dependent effects.",
        "source": "model refinement based on ecological feedback
",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Non-linear
amplification of outbreak decline dynamics",
        "processed": true
    },
    {
        "parameter": "predation_efficiency_exponent",
        "value": 1.0,
```

"description": "Exponent for non-linear scaling of coral predation efficiency. Values >1 amplify the impact of coral cover, capturing threshold dynamics in COTS predation.",
        "source": "model refinement with ecological feedback",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Modulates the non-linear impact of coral cover on COTS predation rates",
        "processed": true
    },
    {
        "parameter": "outbreak_shape",
        "value": 2.0,
        "description": "Exponent controlling non-linear outbreak trigger sensitivity. Values > 1 increase the sharpness of outbreak onset.",
        "source": "model refinement with ecological feedback",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Reflects the sensitivity of outbreak dynamics to surpassing the critical threshold, leading to a rapid outbreak once exceeded",
        "processed": true
    },
    {
        "parameter": "extreme_outbreak_modifier",
        "value": 0.2,
        "description": "Additional multiplier on outbreak sharpness when environmental conditions are extreme, capturing non-linear amplification of outbreak triggers under atypical conditions.",
        "source": "model refinement",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Multiplier for amplifying outbreak trigger sensitivity during extreme environmental conditions",
        "processed": true
    },
    {
        "parameter": "coral_saturation_coefficient",
        "value": 0.1,

```
            "description": "Coefficient for the quadratic term in
    the saturating function of coral availability. This parameter
    captures the non-linear diminishing returns of increased coral
    cover on COTS resource uptake.",
            "source": "model refinement with ecological feedback",
            "import_type": "PARAMETER",
            "priority": 2,
            "enhanced_semantic_description": "Non-linear modulation
    of resource limitation by coral availability",
            "processed": true
        },
        {
            "parameter": "outbreak_hysteresis",
            "value": 0.1,
            "description": "Hysteresis effect in outbreak dynamics
    to capture delayed decline post-outbreak by maintaining outbreak
    conditions longer.",
            "source": "model refinement incorporating delayed
    feedback mechanisms",
            "import_type": "PARAMETER",
            "priority": 2,
            "enhanced_semantic_description": "Captures persistence
    in outbreak conditions due to inertia in system response",
            "processed": true
        },
        {
            "parameter": "outbreak_nonlinearity",
            "value": 0.1,
            "description": "Additional non-linear amplification
    factor in the outbreak trigger. Captures accelerated outbreak
    initiation at high COTS densities.",
            "source": "model improvement",
            "import_type": "PARAMETER",
            "priority": 2,
            "enhanced_semantic_description": "Non-linear
    acceleration factor for outbreak trigger function modeling abrupt
     outbreak dynamics",
            "processed": true
        }
    ]
}
```

S7.5 gpt 4.1 Model

This model achieved an objective value of 0.3488.

S7.5.1 Model Intention

```
This model simulates the population dynamics of Crown-of-Thorns
   starfish (COTS) outbreaks on the Great Barrier Reef, with
   explicit representation of their interactions with fast-growing (
   Acropora spp.) and slow-growing (Faviidae and Porites spp.) coral
    communities. The model captures the characteristic boom-bust
   cycles of COTS by incorporating resource limitation, selective
   predation, and environmental drivers (e.g., sea-surface
   temperature, larval immigration). COTS population growth is
   regulated by coral availability (with saturating functional
   response), environmental conditions, and density-dependent
   feedbacks. Outbreaks are triggered by episodic increases in
   larval supply or favorable environmental conditions, and
   sustained by high coral cover, but collapse as coral is depleted
   or other limiting factors intensify. The model predicts the
   timing, magnitude, and duration of COTS outbreaks, as well as the
    resulting impacts on coral community structure, by simulating
   annual time steps and accounting for both direct and indirect
   ecological feedbacks.
```

S7.5.2 Model Implementation

```
#include <TMB.hpp>

// 1. Model equations are described at the end of this file.

template<class Type>
Type objective_function<Type>::operator() ()
{
  // --- DATA INPUTS ---
  DATA_VECTOR(Year); // Observation year (integer)
  DATA_VECTOR(cots_dat); // Observed adult COTS density (indiv/m2)
  DATA_VECTOR(fast_dat); // Observed fast-growing coral cover (%) (
    Acropora)
  DATA_VECTOR(slow_dat); // Observed slow-growing coral cover (%) (
    Faviidae, Porites)
  DATA_VECTOR(sst_dat); // Sea-surface temperature (deg C)
```

```
DATA_VECTOR(cotsimm_dat); // COTS larval immigration (indiv/m2/
 year)

int n = Year.size();

// --- PARAMETERS ---
PARAMETER(log_r_cots); // log intrinsic COTS growth rate (year^-1)
PARAMETER(log_K_cots); // log COTS carrying capacity (indiv/m2)
PARAMETER(log_alpha_fast); // log COTS attack rate on fast coral (
 m2/indiv/year)
PARAMETER(log_alpha_slow); // log COTS attack rate on slow coral (
 m2/indiv/year)
PARAMETER(log_e_fast); // log assimilation efficiency from fast
 coral (unitless)
PARAMETER(log_e_slow); // log assimilation efficiency from slow
 coral (unitless)
PARAMETER(log_m_cots); // log natural mortality rate of COTS (year
 ^-1)
PARAMETER(logit_theta_sst); // logit SST sensitivity (unitless,
 0-1)
PARAMETER(log_sigma_cots); // log SD for COTS obs (lognormal)
PARAMETER(log_sigma_fast); // log SD for fast coral obs (lognormal
 )
PARAMETER(log_sigma_slow); // log SD for slow coral obs (lognormal
 )
PARAMETER(log_r_fast); // log growth rate of fast coral (year^-1)
PARAMETER(log_r_slow); // log growth rate of slow coral (year^-1)
PARAMETER(log_K_fast); // log carrying capacity fast coral (%)
PARAMETER(log_K_slow); // log carrying capacity slow coral (%)
PARAMETER(log_m_fast); // log background mortality fast coral (
 year^-1)
PARAMETER(log_m_slow); // log background mortality slow coral (
 year^-1)
PARAMETER(logit_phi_outbreak); // logit outbreak threshold (
 unitless, 0-1)
PARAMETER(log_immig_scale); // log scale for larval immigration
 effect

// --- TRANSFORM PARAMETERS ---
Type r_cots = exp(log_r_cots); // COTS intrinsic growth rate
Type K_cots = exp(log_K_cots); // COTS carrying capacity
Type alpha_fast = exp(log_alpha_fast); // COTS attack rate on fast
  coral
```

```
Type alpha_slow = exp(log_alpha_slow); // COTS attack rate on slow
   coral
Type e_fast = exp(log_e_fast); // Assimilation efficiency fast
  coral
Type e_slow = exp(log_e_slow); // Assimilation efficiency slow
  coral
Type m_cots = exp(log_m_cots); // COTS natural mortality
Type theta_sst = Type(1)/(Type(1)+exp(-logit_theta_sst)); // SST
  sensitivity [0,1]
Type sigma_cots = exp(log_sigma_cots); // SD for COTS obs
Type sigma_fast = exp(log_sigma_fast); // SD for fast coral obs
Type sigma_slow = exp(log_sigma_slow); // SD for slow coral obs
Type r_fast = exp(log_r_fast); // Fast coral growth rate
Type r_slow = exp(log_r_slow); // Slow coral growth rate
Type K_fast = exp(log_K_fast); // Fast coral carrying capacity
Type K_slow = exp(log_K_slow); // Slow coral carrying capacity
Type m_fast = exp(log_m_fast); // Fast coral background mortality
Type m_slow = exp(log_m_slow); // Slow coral background mortality
Type phi_outbreak = Type(1)/(Type(1)+exp(-logit_phi_outbreak)); //
   Outbreak threshold [0,1]
Type immig_scale = exp(log_immig_scale); // Immigration effect
  scale

// --- INITIAL STATES ---
Type cots_prev = cots_dat(0); // Initial COTS density (indiv/m2)
Type fast_prev = fast_dat(0); // Initial fast coral cover (%)
Type slow_prev = slow_dat(0); // Initial slow coral cover (%)

// --- SMALL CONSTANTS FOR NUMERICAL STABILITY ---
Type eps = Type(1e-8);

// --- OUTPUT VECTORS ---
vector<Type> cots_pred(n);
vector<Type> fast_pred(n);
vector<Type> slow_pred(n);

// --- PROCESS MODEL ---
for(int t=1; t<n; t++){
  // 1. Resource limitation: total available coral (sum of fast
  and slow)
  Type total_coral_prev = fast_prev + slow_prev + eps;
```

```
// 2. Functional response: COTS predation on coral (Holling Type
 II)
Type pred_fast = alpha_fast * cots_prev * fast_prev / (fast_prev
 + slow_prev + eps); // predation on fast coral
Type pred_slow = alpha_slow * cots_prev * slow_prev / (fast_prev
 + slow_prev + eps); // predation on slow coral

// 3. COTS population growth (logistic, modified by coral
availability and SST)
Type coral_effect = (fast_prev * e_fast + slow_prev * e_slow) /
(K_fast * e_fast + K_slow * e_slow + eps); // resource effect
(0-1)
Type sst_effect = 1.0 + theta_sst * (sst_dat(t) - 27.0); // SST
modifies growth (centered at 27C)
Type immig_effect = immig_scale * cotsimm_dat(t); // immigration
 pulse

// Outbreak trigger: smooth threshold on coral_effect
Type outbreak_boost = 1.0 + phi_outbreak * (coral_effect - 0.5);

Type cots_growth = r_cots * cots_prev * (1.0 - cots_prev / (
K_cots + eps)) * coral_effect * sst_effect * outbreak_boost;
Type cots_mortality = m_cots * cots_prev;

Type cots_next = cots_prev + cots_growth - cots_mortality +
immig_effect;
cots_next = CppAD::CondExpGt(cots_next, eps, cots_next, eps); //
 Bound to >= eps

// 4. Coral dynamics (logistic growth minus COTS predation and
background mortality)
Type fast_growth = r_fast * fast_prev * (1.0 - fast_prev / (
K_fast + eps));
Type fast_mortality = m_fast * fast_prev;
Type fast_next = fast_prev + fast_growth - pred_fast -
fast_mortality;
fast_next = CppAD::CondExpGt(fast_next, eps, fast_next, eps);

Type slow_growth = r_slow * slow_prev * (1.0 - slow_prev / (
K_slow + eps));
Type slow_mortality = m_slow * slow_prev;
Type slow_next = slow_prev + slow_growth - pred_slow -
slow_mortality;
```

```
  slow_next = CppAD::CondExpGt(slow_next, eps, slow_next, eps);

  // Store predictions
  cots_pred(t) = cots_next;
  fast_pred(t) = fast_next;
  slow_pred(t) = slow_next;

  // Update for next step
  cots_prev = cots_next;
  fast_prev = fast_next;
  slow_prev = slow_next;
}

// Set initial predictions to observed initial values
cots_pred(0) = cots_dat(0);
fast_pred(0) = fast_dat(0);
slow_pred(0) = slow_dat(0);

// --- LIKELIHOOD ---
Type nll = 0.0;
for(int t=0; t<n; t++){
  // Lognormal likelihood for strictly positive data
  nll -= dnorm(log(cots_dat(t) + eps), log(cots_pred(t) + eps),
 sigma_cots + eps, true);
  nll -= dnorm(log(fast_dat(t) + eps), log(fast_pred(t) + eps),
 sigma_fast + eps, true);
  nll -= dnorm(log(slow_dat(t) + eps), log(slow_pred(t) + eps),
 sigma_slow + eps, true);
}

// --- REPORTING ---
REPORT(cots_pred);
REPORT(fast_pred);
REPORT(slow_pred);

// --- EQUATION DESCRIPTIONS ---
/*
1. COTS predation on coral: Holling Type II functional response,
 partitioned by coral type.
2. COTS population growth: Logistic, modified by coral
 availability, SST, and outbreak threshold.
3. Coral growth: Logistic, minus COTS predation and background
 mortality.
```

```
  4. Outbreaks triggered by high coral cover and/or larval
    immigration.
  5. All rates and effects are bounded and smoothed for numerical
    stability.
  */

  return nll;
}
```

S7.5.3 Model Parameters


```
{
    "parameters": [
        {
            "parameter": "log_r_cots",
            "value": -0.7,
            "description": "Log 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://www.researchgate.net/figure/Intrinsic-
    growth-rates-following-different-disturbance-types-The-intrinsic-
    growth-rate-r_fig7_50597254",
                "https://link.springer.com/article/10.1007/s00338
    -024-02560-2",
                "https://www.sciencedirect.com/science/article/pii/
    S0304380023001746"
            ],
            "processed": true
        },
        {
            "parameter": "log_K_cots",
            "value": -0.5,
            "description": "Log carrying capacity for COTS (indiv/m2
    )",
            "source": "expert opinion",
            "import_type": "PARAMETER",
            "priority": 2,
```

```
        "enhanced_semantic_description": "Maximum sustainable
Crown of Thorns starfish population density",
        "processed": true
    },
    {
        "parameter": "log_alpha_fast",
        "value": -2.0,
        "description": "Log attack rate of COTS on fast coral (
m2/indiv/year)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Predation rate of COTS
 on fast-growing branching coral species",
        "citations": [
            "https://www.sciencedirect.com/science/article/pii/
S0048969724028389",
            "https://www.sciencedirect.com/science/article/pii/
S0141113624003167",
            "https://link.springer.com/article/10.1007/s00338
-024-02560-2"
        ],
        "processed": true
    },
    {
        "parameter": "log_alpha_slow",
        "value": -2.5,
        "description": "Log attack rate of COTS on slow coral (
m2/indiv/year)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Predation rate of COTS
 on slow-growing massive coral species",
        "citations": [
            "https://www.sciencedirect.com/science/article/pii/
S0048969724028389",
            "https://link.springer.com/article/10.1007/s00338
-024-02560-2",
            "https://www.int-res.com/articles/theme/m512p167.pdf
"
        ],
        "processed": true
```

```
    },
    {
        "parameter": "log_e_fast",
        "value": -0.1,
        "description": "Log assimilation efficiency from fast
coral (unitless)",
        "source": "expert opinion",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Nutritional efficiency
 of fast-growing coral for COTS",
        "processed": true
    },
    {
        "parameter": "log_e_slow",
        "value": -0.2,
        "description": "Log assimilation efficiency from slow
coral (unitless)",
        "source": "expert opinion",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Nutritional efficiency
 of slow-growing coral for COTS",
        "processed": true
    },
    {
        "parameter": "log_m_cots",
        "value": -1.0,
        "description": "Log natural mortality rate of COTS (year
^-1)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "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_popula
",
            "https://pmc.ncbi.nlm.nih.gov/articles/PMC9023020/"
```

```
        ],
        "processed": true,
        "found_value": 2.25,
        "found_min": 2.25,
        "found_max": 2.25
    },
    {
        "parameter": "logit_theta_sst",
        "value": 0.0,
        "description": "Logit SST sensitivity (unitless, 0-1)",
        "source": "expert opinion",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Sea surface
temperature sensitivity for COTS population dynamics",
        "processed": true
    },
    {
        "parameter": "log_sigma_cots",
        "value": -1.0,
        "description": "Log SD for COTS obs (lognormal)",
        "source": "initial estimate",
        "import_type": "PARAMETER",
        "priority": 4,
        "enhanced_semantic_description": "Measurement
uncertainty in Crown of Thorns starfish population estimates",
        "processed": true
    },
    {
        "parameter": "log_sigma_fast",
        "value": -1.0,
        "description": "Log SD for fast coral obs (lognormal)",
        "source": "initial estimate",
        "import_type": "PARAMETER",
        "priority": 4,
        "enhanced_semantic_description": "Measurement
uncertainty in fast-growing coral cover estimates",
        "processed": true
    },
    {
        "parameter": "log_sigma_slow",
        "value": -1.0,
        "description": "Log SD for slow coral obs (lognormal)",
```

```
        "source": "initial estimate",
        "import_type": "PARAMETER",
        "priority": 4,
        "enhanced_semantic_description": "Measurement
uncertainty in slow-growing coral cover estimates",
        "processed": true
    },
    {
        "parameter": "log_r_fast",
        "value": -0.5,
        "description": "Log growth rate of fast coral (year^-1)
",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Growth rate of fast-
growing branching coral species",
        "citations": [
            "https://coralcavern.com/how-fast-does-coral-grow-in
-colonies/",
            "https://www.nature.com/articles/s41598
-017-03085-1",
            "https://www.sciencedirect.com/science/article/abs/
pii/S0025322719302452"
        ],
        "processed": true,
        "found_value": 3.77,
        "found_min": 2.3,
        "found_max": 4.61
    },
    {
        "parameter": "log_r_slow",
        "value": -1.0,
        "description": "Log growth rate of slow coral (year^-1)
",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Growth rate of slow-
growing massive coral species",
        "citations": [
            "https://www.sciencedirect.com/science/article/abs/
pii/S0025322719302452",
```

"https://www.sciencedirect.com/science/article/pii/
S0925857418303094",
            "https://www.livingoceansfoundation.org/wp-content/
uploads/2015/04/U9-Coral-Growth-Background.pdf"
        ],
        "processed": true,
        "found_value": -0.35,
        "found_min": -0.7,
        "found_max": 0.0
    },
    {
        "parameter": "log_K_fast",
        "value": 3.5,
        "description": "Log carrying capacity for fast coral (%)
",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Maximum sustainable
cover for fast-growing coral species",
        "citations": [
            "https://www.sciencedirect.com/science/article/pii/
S0025326X23001522",
            "https://www.straitstimes.com/asia/australianz/great
-barrier-reef-shows-big-increase-in-coral-cover-but-future-
uncertain-report",
            "https://www.nature.com/articles/s41467-022-30234-6"
        ],
        "processed": true
    },
    {
        "parameter": "log_K_slow",
        "value": 3.0,
        "description": "Log carrying capacity for slow coral (%)
",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Maximum sustainable
cover for slow-growing coral species",
        "citations": [
            "https://www.sciencedirect.com/science/article/pii/
S0025326X23001522",

```
            "https://link.springer.com/article/10.1007/s00338
-024-02560-2",
            "https://cosmosmagazine.com/earth/great-barrier-reef
-coral-cover/"
        ],
        "processed": true,
        "found_value": 52.5,
        "found_min": 40.0,
        "found_max": 65.0
    },
    {
        "parameter": "log_m_fast",
        "value": -2.0,
        "description": "Log background mortality rate of fast
coral (year^-1)",
        "source": "expert opinion",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Background mortality
rate for fast-growing coral species",
        "processed": true
    },
    {
        "parameter": "log_m_slow",
        "value": -2.0,
        "description": "Log background mortality rate of slow
coral (year^-1)",
        "source": "expert opinion",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Background mortality
rate for slow-growing coral species",
        "processed": true
    },
    {
        "parameter": "logit_phi_outbreak",
        "value": 0.0,
        "description": "Logit outbreak threshold (unitless, 0-1)
",
        "source": "expert opinion",
        "import_type": "PARAMETER",
        "priority": 2,
```

```
            "enhanced_semantic_description": "Threshold probability
    for Crown of Thorns starfish population outbreak",
            "processed": true
        },
        {
            "parameter": "log_immig_scale",
            "value": -1.0,
            "description": "Log scale for larval immigration effect
    ",
            "source": "expert opinion",
            "import_type": "PARAMETER",
            "priority": 2,
            "enhanced_semantic_description": "Magnitude of larval
    immigration impact on COTS population",
            "processed": true
        }
    ]
}
```

S8 Best Out-of-Sample Test Model

This section presents the best performing out-of-sample test model.

S8.1 Model Intention

```
Assessment:
- The current model includes resource limitation, environmental
   drivers, and feedbacks between COTS and coral, but does not fully
    capture the episodic, explosive nature of COTS outbreaks.
- The outbreak threshold is a smooth function of coral cover, but
   there is no explicit positive feedback in COTS recruitment based
   on previous COTS abundance, which is a key ecological process in
   outbreak dynamics.
- Field studies indicate that high COTS densities can amplify
   recruitment in subsequent years, leading to rapid population
   booms.

Improvement:
- Add a lagged positive feedback mechanism to COTS recruitment,
   where high COTS abundance in the previous year increases
   recruitment in the current year.
- This is implemented as a saturating Hill-type function of previous
    COTS abundance, controlled by a new parameter (phi_cots).
- This change is ecologically justified as it allows the model to
   generate sharper, more episodic outbreaks, better matching
   observed boom-bust cycles and the timing/magnitude of population
   explosions.
```

S8.2 Model Implementation

```
#include <TMB.hpp>

// 1. Model equations describe the coupled dynamics of COTS, fast
   coral, and slow coral.
// 2. Resource limitation is modeled with saturating and threshold
   functions.
// 3. Environmental drivers (SST, larval immigration) modulate COTS
   recruitment.
// 4. Feedbacks: COTS reduce coral, coral depletion limits COTS,
   coral recovers after COTS decline.
```

```
// 5. All _pred variables are reported and correspond to _dat
   observations.
// 6. No current time step values of _dat variables are used in
   predictions (no data leakage).

template<class Type>
Type objective_function<Type>::operator() ()
{
  // --- DATA INPUTS ---
  DATA_VECTOR(Year); // Observation year
  DATA_VECTOR(cots_dat); // Adult COTS abundance (indiv/m2)
  DATA_VECTOR(fast_dat); // Fast coral cover (%)
  DATA_VECTOR(slow_dat); // Slow coral cover (%)
  DATA_VECTOR(sst_dat); // Sea-surface temperature (deg C)
  DATA_VECTOR(cotsimm_dat); // COTS larval immigration (indiv/m2/
    year)

  int n = Year.size();

  // --- GUARD AGAINST EMPTY INPUT ---
  if(n == 0) {
    // Return large penalty if no data
    return Type(1e10);
  }

  // --- PARAMETERS ---
  PARAMETER(log_r_cots); // log intrinsic COTS recruitment rate (
    year^-1)
  PARAMETER(log_K_cots); // log COTS carrying capacity (indiv/m2)
  PARAMETER(log_K_cots_half); // log coral cover at which COTS K is
    half-maximal
  PARAMETER(log_alpha_fast); // log COTS predation rate on fast
    coral (m2/indiv/year)
  PARAMETER(log_alpha_slow); // log COTS predation rate on slow
    coral (m2/indiv/year)
  PARAMETER(log_r_fast); // log fast coral regrowth rate (year^-1)
  PARAMETER(log_r_slow); // log slow coral regrowth rate (year^-1)
  PARAMETER(log_K_fast); // log fast coral max cover (%)
  PARAMETER(log_K_slow); // log slow coral max cover (%)
  PARAMETER(log_beta_sst); // log SST effect on COTS recruitment (
    unitless)
  PARAMETER(log_imm_eff); // log efficiency of larval immigration (
    unitless)
```

```
PARAMETER(log_sigma_cots); // log obs SD for COTS (lognormal)
PARAMETER(log_sigma_fast); // log obs SD for fast coral (lognormal
 )
PARAMETER(log_sigma_slow); // log obs SD for slow coral (lognormal
 )

// --- TRANSFORM PARAMETERS ---
Type r_cots = exp(log_r_cots); // Intrinsic COTS recruitment rate
 (year^-1)
Type K_cots = exp(log_K_cots); // COTS carrying capacity (indiv/m2
 )
Type K_cots_half = exp(log_K_cots_half); // Coral cover at which
 COTS K is half-maximal
Type alpha_fast = exp(log_alpha_fast); // COTS predation rate on
 fast coral (m2/indiv/year)
Type alpha_slow = exp(log_alpha_slow); // COTS predation rate on
 slow coral (m2/indiv/year)
Type r_fast = exp(log_r_fast); // Fast coral regrowth rate (year
 ^-1)
Type r_slow = exp(log_r_slow); // Slow coral regrowth rate (year
 ^-1)
Type K_fast = exp(log_K_fast); // Fast coral max cover (%)
Type K_slow = exp(log_K_slow); // Slow coral max cover (%)
Type beta_sst = exp(log_beta_sst); // SST effect on COTS
 recruitment (unitless)
Type imm_eff = exp(log_imm_eff); // Efficiency of larval
 immigration (unitless)
Type sigma_cots = exp(log_sigma_cots); // Obs SD for COTS (
 lognormal)
Type sigma_fast = exp(log_sigma_fast); // Obs SD for fast coral (
 lognormal)
Type sigma_slow = exp(log_sigma_slow); // Obs SD for slow coral (
 lognormal)
PARAMETER(log_gamma_cots); // log COTS interference strength (
 density-dependent reduction in per capita predation)
Type gamma_cots = exp(log_gamma_cots); // COTS interference
 strength

PARAMETER(log_phi_cots); // log positive feedback strength in COTS
  recruitment (lagged autocatalytic effect)
Type phi_cots = exp(log_phi_cots); // Positive feedback strength

// --- INITIAL STATES ---
```

```
Type cots_prev = cots_dat(0); // Initial COTS abundance (indiv/m2)
Type fast_prev = fast_dat(0); // Initial fast coral cover (%)
Type slow_prev = slow_dat(0); // Initial slow coral cover (%)

// --- OUTPUT VECTORS ---
vector<Type> cots_pred(n);
vector<Type> fast_pred(n);
vector<Type> slow_pred(n);

// --- SMALL CONSTANT FOR NUMERICAL STABILITY ---
Type eps = Type(1e-8);

// --- INITIALIZE PREDICTIONS ---
cots_pred(0) = cots_prev;
fast_pred(0) = fast_prev;
slow_pred(0) = slow_prev;

// --- PROCESS MODEL ---
for(int t=1; t<n; t++) {
  // 1. COTS recruitment: logistic growth, modulated by SST,
  larval immigration, and lagged positive feedback
  Type env_mod = 1 + beta_sst * (sst_dat(t-1) - Type(27.0)); //
  SST effect (centered at 27C)
  Type immig = imm_eff * cotsimm_dat(t-1); // Immigration effect

  // Resource limitation: carrying capacity depends on coral cover
   (saturating, Michaelis-Menten)
  Type coral_sum = fast_prev + slow_prev + eps;
  Type K_cots_eff = K_cots * (coral_sum/(K_cots_half + coral_sum +
   eps)); // COTS K saturates with total coral

  // Outbreak threshold: smooth sigmoid on COTS recruitment (
  triggers outbreak when coral is high)
  Type outbreak_mod = 1/(1 + exp(-5*(coral_sum - 10))); //
  Outbreak more likely if coral >10%

  // Lagged positive feedback in COTS recruitment (Hill-type
  function of previous COTS abundance)
  Type feedback_mod = 1 + phi_cots * cots_prev / (1 + phi_cots *
  cots_prev);

  // COTS predation on corals (Type II functional response) with
  density-dependent interference
```

```
  Type interference = exp(-gamma_cots * cots_prev); // Reduces per
   capita predation at high COTS density
  Type pred_fast = alpha_fast * cots_prev * fast_prev / (fast_prev
   + Type(5.0) + eps) * interference; // Fast coral eaten
  Type pred_slow = alpha_slow * cots_prev * slow_prev / (slow_prev
   + Type(10.0) + eps) * interference; // Slow coral eaten

  // COTS population update
  Type cots_growth = r_cots * cots_prev * (1 - cots_prev/(
 K_cots_eff+eps)) * env_mod * outbreak_mod * feedback_mod;
  Type cots_next = cots_prev + cots_growth + immig - pred_fast
 *0.05 - pred_slow*0.02; // Small mortality from feeding
 inefficiency

  // Bound COTS to positive values
  cots_next = CppAD::CondExpGt(cots_next, eps, cots_next, eps);

  // Fast coral update: logistic regrowth minus COTS predation
  Type fast_growth = r_fast * fast_prev * (1 - fast_prev/(K_fast+
 eps));
  Type fast_next = fast_prev + fast_growth - pred_fast;

  fast_next = CppAD::CondExpGt(fast_next, eps, fast_next, eps);

  // Slow coral update: logistic regrowth minus COTS predation
  Type slow_growth = r_slow * slow_prev * (1 - slow_prev/(K_slow+
 eps));
  Type slow_next = slow_prev + slow_growth - pred_slow;

  slow_next = CppAD::CondExpGt(slow_next, eps, slow_next, eps);

  // Save predictions
  cots_pred(t) = cots_next;
  fast_pred(t) = fast_next;
  slow_pred(t) = slow_next;

  // Advance state
  cots_prev = cots_next;
  fast_prev = fast_next;
  slow_prev = slow_next;
}

// --- LIKELIHOOD ---
```

```
  Type nll = 0.0;
  for(int t=0; t<n; t++) {
    // Lognormal likelihood, fixed minimum SD for stability
    Type sd_cots = sqrt(sigma_cots*sigma_cots + eps);
    Type sd_fast = sqrt(sigma_fast*sigma_fast + eps);
    Type sd_slow = sqrt(sigma_slow*sigma_slow + eps);

    nll -= dnorm(log(cots_dat(t)+eps), log(cots_pred(t)+eps),
   sd_cots, true);
    nll -= dnorm(log(fast_dat(t)+eps), log(fast_pred(t)+eps),
   sd_fast, true);
    nll -= dnorm(log(slow_dat(t)+eps), log(slow_pred(t)+eps),
   sd_slow, true);
  }

  // --- SMOOTH PENALTIES FOR PARAMETER BOUNDS ---
  // Example: penalize negative growth rates, unreasonably high K,
   etc.
  nll += pow(CppAD::CondExpLt(r_cots, Type(0.01), r_cots-Type(0.01),
    Type(0)), 2) * 10.0;
  nll += pow(CppAD::CondExpGt(K_cots, Type(10.0), K_cots-Type(10.0),
    Type(0)), 2) * 10.0;
  nll += pow(CppAD::CondExpLt(r_fast, Type(0.01), r_fast-Type(0.01),
    Type(0)), 2) * 10.0;
  nll += pow(CppAD::CondExpLt(r_slow, Type(0.01), r_slow-Type(0.01),
    Type(0)), 2) * 10.0;

  // --- REPORTING ---
  REPORT(cots_pred); // Predicted COTS abundance (indiv/m2)
  REPORT(fast_pred); // Predicted fast coral cover (%)
  REPORT(slow_pred); // Predicted slow coral cover (%)

  return nll;
}
```

S8.3 Model Parameters


```
{
    "parameters": [
        {
            "parameter": "log_r_cots",
            "value": -0.7,
```

```
        "description": "Log intrinsic COTS recruitment rate (
year^-1)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 1,
        "enhanced_semantic_description": "Crown-of-thorns
starfish population growth potential rate",
        "citations": [
            "https://www.sciencedirect.com/science/article/pii/
S0048969724054329",
            "https://pmc.ncbi.nlm.nih.gov/articles/PMC9023020/",
            "https://www.reefresilience.org/pdf/COTS_Nov2003.pdf
"
        ],
        "processed": true
    },
    {
        "parameter": "log_K_cots",
        "value": -1.0,
        "description": "Log COTS carrying capacity (indiv/m2),
scales with coral cover",
        "source": "expert opinion",
        "import_type": "PARAMETER",
        "priority": 1,
        "enhanced_semantic_description": "Maximum sustainable
Crown-of-thorns starfish population density",
        "processed": true
    },
    {
        "parameter": "log_K_cots_half",
        "value": 2.5,
        "description": "Log coral cover at which COTS carrying
capacity is half its maximum (saturating resource limitation)",
        "source": "ecological reasoning",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Coral cover at which
COTS carrying capacity is half-maximal; controls saturation of
resource limitation",
        "processed": true
    },
    {
        "parameter": "log_alpha_fast",
```

```
        "value": -2.0,
        "description": "Log COTS predation rate on fast coral (
m2/indiv/year)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Predation impact of
COTS on fast-growing coral species",
        "citations": [
            "https://www.sciencedirect.com/science/article/pii/
S0048969724028389",
            "https://link.springer.com/article/10.1007/s00338
-024-02560-2",
            "https://reefbites.com/2023/04/26/crown-of-thorns-
starfish-cots-the-complicated-story-of-a-natural-predator-on-
coral-reefs/"
        ],
        "processed": true
    },
    {
        "parameter": "log_alpha_slow",
        "value": -3.0,
        "description": "Log COTS predation rate on slow coral (
m2/indiv/year)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Predation impact of
COTS on slow-growing coral species",
        "citations": [
            "https://www.sciencedirect.com/science/article/pii/
S0048969724028389",
            "https://www.nature.com/articles/s41467
-021-26786-8",
            "https://link.springer.com/article/10.1007/s00338
-024-02560-2"
        ],
        "processed": true
    },
    {
        "parameter": "log_r_fast",
        "value": -0.5,
        "description": "Log fast coral regrowth rate (year^-1)",
```

```
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Rapid coral recovery
and regeneration potential rate",
        "citations": [
            "https://www.sciencedirect.com/science/article/pii/
S0960982224001519",
            "https://www.cell.com/current-biology/fulltext/S0960
-9822(24)00151-9?rss=yes",
            "https://pmc.ncbi.nlm.nih.gov/articles/PMC9331011/"
        ],
        "processed": true
    },
    {
        "parameter": "log_r_slow",
        "value": -1.0,
        "description": "Log slow coral regrowth rate (year^-1)",
        "source": "literature",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Slow coral recovery
and regeneration potential rate",
        "citations": [
            "https://esajournals.onlinelibrary.wiley.com/doi
/10.1002/ecy.4510",
            "https://esajournals.onlinelibrary.wiley.com/doi
/10.1002/ecs2.4915",
            "https://www.sciencedirect.com/science/article/abs/
pii/S0025326X15001940"
        ],
        "processed": true
    },
    {
        "parameter": "log_K_fast",
        "value": 3.0,
        "description": "Log fast coral max cover (%)",
        "source": "expert opinion",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Maximum sustainable
coverage for fast-growing coral species",
        "processed": true
```

```
    },
    {
        "parameter": "log_K_slow",
        "value": 3.0,
        "description": "Log slow coral max cover (%)",
        "source": "expert opinion",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Maximum sustainable
coverage for slow-growing coral species",
        "processed": true
    },
    {
        "parameter": "log_beta_sst",
        "value": -2.0,
        "description": "Log SST effect on COTS recruitment (
unitless)",
        "source": "initial estimate",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Sea surface
temperature influence on COTS population dynamics",
        "processed": true
    },
    {
        "parameter": "log_imm_eff",
        "value": -2.0,
        "description": "Log efficiency of larval immigration (
unitless)",
        "source": "initial estimate",
        "import_type": "PARAMETER",
        "priority": 3,
        "enhanced_semantic_description": "Larval immigration
effectiveness in COTS population expansion",
        "processed": true
    },
    {
        "parameter": "log_sigma_cots",
        "value": -1.0,
        "description": "Log obs SD for COTS (lognormal)",
        "source": "initial estimate",
        "import_type": "PARAMETER",
        "priority": 1,
```

```
        "enhanced_semantic_description": "Measurement
uncertainty for Crown-of-thorns starfish population",
        "processed": true
    },
    {
        "parameter": "log_sigma_fast",
        "value": -1.0,
        "description": "Log obs SD for fast coral (lognormal)",
        "source": "initial estimate",
        "import_type": "PARAMETER",
        "priority": 1,
        "enhanced_semantic_description": "Measurement
uncertainty for fast-growing coral species coverage",
        "processed": true
    },
    {
        "parameter": "log_sigma_slow",
        "value": -1.0,
        "description": "Log obs SD for slow coral (lognormal)",
        "source": "initial estimate",
        "import_type": "PARAMETER",
        "priority": 1,
        "enhanced_semantic_description": "Measurement
uncertainty for slow-growing coral species coverage",
        "processed": true
    },
    {
        "parameter": "log_gamma_cots",
        "value": -2.0,
        "description": "Log COTS interference strength (density-
dependent reduction in per capita predation)",
        "source": "ecological reasoning",
        "import_type": "PARAMETER",
        "priority": 2,
        "enhanced_semantic_description": "Strength of
intraspecific interference among COTS; higher values reduce per
capita predation at high COTS densities",
        "processed": true
    },
    {
        "parameter": "log_phi_cots",
        "value": -2.0,
```

```
            "description": "Log positive feedback strength in COTS
    recruitment (lagged autocatalytic effect)",
            "source": "ecological reasoning",
            "import_type": "PARAMETER",
            "priority": 2,
            "enhanced_semantic_description": "Strength of lagged
    positive feedback in COTS recruitment; higher values increase the
     likelihood and magnitude of outbreak events when previous COTS
    abundance is high",
            "processed": true
        }
    ]
}
```