AI-ASSISTED MODEL-DRIVEN EPIDEMIOLOGY

FATEMEH SEYEDDABBAGHI

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF COMPUTER SCIENCE

GRADUATE PROGRAM IN ELECTRICAL ENGINEERING & COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

AUGUST 2026

© Fatemeh Seyeddabbaghi, 2026

Abstract

Compartmental models represent infectious disease dynamics, but converting published papers into formal, executable models remains difficult: model structure is scattered across prose, equations, tables, and supplementary material, and reconstructing it requires expert interpretation. This thesis develops an AI-assisted model-driven pipeline built around *EpiMDE*, a model-driven engineering platform providing a metamodel for compartments, flows, stratification, and parameters. The pipeline is evaluated on a thirty-paper benchmark spanning ten diseases. Phase 1 establishes hand-authored reference models. Phase 2 extracts draft models from PDFs using metamodel-grounded LLM prompting across three backends; per-paper best-provider selection yields mean compartment recall 0.80 and flow recall 0.57. Phase 3 applies retrieval-augmented completion, raising these to 0.96 and 0.84 and mean composite completeness from 60.7% to 91.2%. Phase 4 quantifies parameter uncertainty using Monte Carlo simulation and sensitivity analysis. A feature-level layer maps each model to seventeen epidemiological features, exposing extraction blind spots and supporting traceability and reproducibility.

Dedication

In solemn memory of those who gave their lives in the pursuit of a free Iran.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor Marios Fokaefs, for his guidance, feedback, and support throughout this research. His advice helped shape the direction of this thesis and supported the development of the methodology, evaluation, and presentation of the work.

I would also like to thank the committee members for serving on my thesis committee and for their contribution to the completion of this thesis.

I am grateful to my collaborators for discussions related to EpiMDE, epidemiological modelling, model-driven engineering, and AI-assisted model extraction. These conversations helped refine the ideas and practical decisions behind this research.

I would like to thank my family and friends for their patience, encouragement, and support during my graduate studies. Their support helped me continue through the challenging stages of this work.

Finally, I thank York University and the Department of Electrical Engineering and Computer Science for providing the academic environment in which this thesis was completed.

Declaration of AI Use

Artificial intelligence tools were used during the preparation of this thesis to support editing, language refinement, formatting assistance, and the drafting of selected explanatory text. In particular, ChatGPT was used to help improve clarity, grammar, structure, and consistency across chapters, and to assist with LaTeX formatting and figure/table presentation.

AI tools were not used as independent sources of scientific evidence. All technical claims, methodological decisions, experimental results, citations, and interpretations remain the responsibility of the author. The generated or revised text was reviewed, edited, and verified by the author before inclusion in the thesis. The use of AI tools in this thesis is disclosed in accordance with IEEE guidance on AI-generated content, which recommends identifying the AI system used and explaining the level of use.

Table of Contents

Abstract ii

Dedication iii

Acknowledgements iv

Declaration of AI Use v

Table of Contents vi

List of Tables xiii

List of Figures xiv

Abbreviations xv

1 Introduction 1

1.1 Context and Motivation 1
1.2 The Case for Model-Driven Engineering 2
1.3 Artificial Intelligence for Model Extraction and Completion 3
1.4 Research Problem and Questions 3
1.5 Contributions 4

2 Background and Related Work 6

2.1 Compartmental Epidemiological Models 6
2.1.1 Foundations 6
2.1.2 The SIR Model 7
2.1.3 The SEIR Model and Extensions 8
2.1.4 Flow Types 8
2.1.5 Challenges of Transcribing Published Models 9
2.2 Model-Driven Engineering 9
2.2.1 Core Principles 9

2.2.2 Metamodels and Domain-Specific Languages . . . 10
2.2.3 MDE in Epidemiology . . . 10
2.3 Large Language Models and Retrieval-Augmented Generation . . . 10
2.3.1 Transformer Architecture . . . 11
2.3.2 LLMs for Structured Extraction . . . 11
2.3.3 Retrieval-Augmented Generation . . . 11
2.4 Related Work . . . 12
2.4.1 AI-Assisted Epidemiological Modeling . . . 12
2.4.2 LLM-Based Extraction from Scientific Documents . . . 13
2.4.3 MDE Tools for Epidemiology and Related Domains . . . 13
2.4.4 Uncertainty Quantification in Epidemic Modeling . . . 13
2.4.5 Reproducibility in Computational Epidemiology . . . 14
2.4.6 Positioning This Thesis . . . 14

3 The EpiMDE Platform 15
3.1 Design Goals . . . 15
3.2 The EpiMDE Metamodel . . . 16
3.2.1 Overview . . . 16
3.2.2 Compartments and Flows . . . 17
3.2.3 Population Stratification . . . 18
3.2.4 External Sources and Sinks . . . 18
3.2.5 The Parameter System . . . 18
3.3 Disease Model Examples . . . 19
3.3.1 Overview of Bundled Models . . . 19
3.3.2 The COVID-19 Model . . . 19
3.3.3 The HIV Model . . . 19
3.4 The EpiMDE IDE . . . 19
3.4.1 Architecture . . . 19
3.4.2 Graphical and Tree Editors . . . 22
3.5 Equation Generation . . . 23
3.5.1 CompartmentalEquationGenerator . . . 23
3.5.2 Python Simulation . . . 23
3.6 The Role of EpiMDE in the AI-Assisted Pipeline . . . 23

4 Phase 1: Baseline Analysis **25**
4.1 Motivation and Goals . . . 25
4.2 Gold-Standard Corpus . . . 26
4.2.1 Corpus Construction . . . 26
4.2.2 Benchmark Layout . . . 27
4.3 Structural Analysis . . . 27
4.3.1 Methodology . . . 27
4.3.2 Results . . . 27
4.4 Gap Analysis: Paper Promises vs. Model Content . . . 28
4.4.1 Defining Paper Promises . . . 28
4.4.2 Promise Extraction Methodology . . . 28
4.4.3 Gap Analysis Results . . . 29
4.5 Parameter Uncertainty Documentation . . . 29
4.5.1 Methodology . . . 29
4.5.2 Key Findings . . . 30
4.6 Sensitivity Analysis . . . 30
4.6.1 Methodology . . . 30
4.6.2 Results . . . 30
4.7 Shared Artifacts for Downstream Phases . . . 31
4.8 Chapter Summary . . . 31

5 Phase 2: Model Extraction from Scientific Literature **32**
5.1 Motivation and Design Goals . . . 32
5.1.1 The Faithfulness Principle . . . 32
5.1.2 Epidemiology Scope Steering . . . 33
5.1.3 Multi-Provider Support . . . 33
5.1.4 Span-Level Traceability . . . 33
5.2 The Seven-Step Pipeline . . . 34
5.2.1 Step 1: PDF Text Extraction . . . 34
5.2.2 Step 2: Paper Promise Extraction . . . 35
5.2.3 Step 3: Entity Extraction via LLM Prompting . . . 35
5.2.4 Step 4: Model Synthesis . . . 36
5.2.5 Step 5: Traceability Linking . . . 36
5.2.6 Step 6: Quality Checks . . . 36
5.2.7 Step 7: Evaluation Against Gold Standard . . . 37

5.3 Benchmark and Evaluation Methodology . . . 37
5.3.1 Benchmark Corpus . . . 37
5.3.2 Gold Standard Construction . . . 37
5.3.3 Evaluation Metrics . . . 38
5.4 LLM Provider Configurations . . . 38
5.5 Results . . . 38
5.5.1 Compartment Recall . . . 39
5.5.2 Flow Recall . . . 39
5.5.3 Best-Per-Disease Upper Bound . . . 39
5.6 Error Analysis . . . 41
5.6.1 Ambiguous Prose . . . 41
5.6.2 Noisy PDF Text . . . 41
5.6.3 Unsupported Inference . . . 41
5.7 Traceability . . . 42
5.8 Chapter Summary . . . 42

6 Phase 3: Gap Detection and Model Completion 43
6.1 Motivation . . . 44
6.1.1 Structural Completeness as a Prerequisite for Simulation . . . 44
6.1.2 Sources of Structural Gaps . . . 45
6.1.3 The Retrieval Augmentation Strategy . . . 45
6.2 Gap Detection . . . 45
6.2.1 Overview . . . 45
6.2.2 Promise-Based Gap Detection . . . 46
6.2.3 Fuzzy Matching for Gap Alignment . . . 46
6.3 Building the Paper Database . . . 47
6.3.1 Database Architecture . . . 47
6.4 Three-Tier Gap Filling . . . 47
6.4.1 Tier 1: Rule-Based Retrieval . . . 48
6.4.2 Tier 2: LLM Inference . . . 48
6.4.3 Tier 3: Explicit Flagging . . . 49
6.5 Validation . . . 49
6.5.1 Numeric Validation . . . 50
6.5.2 Structural Validation . . . 50
6.5.3 Composite Completeness Metric . . . 50

6.6 Component Contribution Analysis . . . 51
6.6.1 Complementarity of the Two Active Tiers . . . 51
6.6.2 Best-Mode Selection . . . 52
6.7 Results . . . 52
6.7.1 Aggregate Improvement . . . 52
6.7.2 Per-Disease Results . . . 52
6.7.3 Composite Completeness . . . 54
6.8 Residual Gaps . . . 54
6.8.1 Disconnected Flows . . . 54
6.8.2 Inconsistent Units . . . 54
6.8.3 Absent from All Sources . . . 55
6.9 Chapter Summary . . . 55

7 Phase 4: Uncertainty Quantification 56
7.1 Rationale for Uncertainty Quantification . . . 56
7.2 Model Selection from Phase 3 . . . 57
7.3 Parameter Distribution Assignment . . . 57
7.3.1 Distribution Families . . . 58
7.3.2 Keyword-to-Distribution Mapping . . . 58
7.3.3 Computing Lognormal Parameters . . . 58
7.4 Monte Carlo Simulation . . . 59
7.4.1 Simulation Protocol . . . 59
7.4.2 Extracted Summary Statistics . . . 59
7.4.3 Monte Carlo Visualization . . . 60
7.5 Sensitivity Analysis . . . 61
7.5.1 One-at-a-Time Perturbation Method . . . 61
7.5.2 Sensitivity Bar Chart Visualisation . . . 64
7.6 Engineering Challenges . . . 65
7.6.1 Zero Point Estimates . . . 65
7.6.2 Negative Parameter Values . . . 66
7.6.3 Inconsistent Initial Conditions . . . 66
7.7 Results . . . 66
7.8 Interpretation and Limitations . . . 67
7.9 Chapter Summary . . . 67

8 Feature-Level Evaluation **69**
8.1 Motivation for Feature-Level Analysis . . . 69
8.2 The EpiFeatureVector . . . 70
8.3 Inference Rules . . . 72
8.3.1 Structure of the Rule Engine . . . 72
8.3.2 Avoiding False Matches . . . 72
8.3.3 Disease-Name Priors . . . 73
8.4 Reference Profiles . . . 73
8.5 Feature Model Diagrams . . . 74
8.6 Text-Grounded Recall . . . 75
8.6.1 Problem: Prior-Added Features . . . 75
8.6.2 Definition of Text-Grounded Recall . . . 75
8.6.3 Practical Impact . . . 76
8.7 Results . . . 76
8.7.1 The Standard-versus-Text-Grounded Gap . . . 76
8.7.2 Disease Difficulty Ranking . . . 77
8.7.3 Systematically Missed Feature Categories . . . 78
8.8 Chapter Summary . . . 78

9 Discussion and Conclusion **79**
9.1 Cross-Phase Synthesis . . . 79
9.1.1 Phase 1: Reference Space Establishment . . . 79
9.1.2 Phase 2: LLM-Based Extraction . . . 80
9.1.3 Phase 3: Completion and Gap Filling . . . 80
9.1.4 Feature-Level Evaluation . . . 80
9.1.5 Phase 4: Uncertainty Quantification . . . 81
9.1.6 The Cumulative Pipeline Progression . . . 81
9.2 Feature-Level Evaluation as a Diagnostic Lens . . . 81
9.3 Answering the Research Questions . . . 82
9.3.1 RQ1: Extraction Quality of LLM-Based Compartmental Model Drafting 82
9.3.2 RQ2: Completion and Residual Gaps . . . 82
9.3.3 RQ3: Uncertainty Propagation and Sensitivity . . . 83
9.3.4 RQ4: Reproducibility and Methodological Transparency . . . 83
9.4 Threats to Validity . . . 84
9.4.1 Construct Validity . . . 84

9.4.2 Internal Validity 84
9.4.3 External Validity 85
9.4.4 Conclusion Validity 85
9.5 Limitations 85
9.6 Future Work 86
9.7 Conclusion 86

A Supplementary Material **88**
A.1 Annotated `.compmodel` XML Reference 88
A.2 Benchmark Corpus 91

Bibliography **94**

List of Tables

3.1 Core EpiMDE metamodel elements and their epidemiological roles. 17

4.1 Ten-disease benchmark corpus used in the evaluation. 26
4.2 Structural characteristics of the ten primary gold-standard models. 28

5.1 Average Phase 2 extraction results by LLM provider. 39
5.2 Per-paper Phase 2 extraction results using the best provider for each paper. . 40

6.1 Composition of the Phase 3 paper database. 48
6.2 Phase 3 component contribution study across the 30 benchmark models. ParamFill is the macro-average fraction of flows whose rate or contact-rate parameter has a numeric non-placeholder expression. 51
6.3 Per-paper recall using the best Phase 2 LLM provider and the gap-selected Phase 3 mode per paper. Completeness follows Equation 6.1. 53

7.1 Parameter distributions used in the Phase 4 uncertainty framework. 59

8.1 Epidemiological features represented in the EpiFeatureVector, grouped by category. 71
8.2 Feature-level evaluation results by disease. 77

A.1 Thirty-paper benchmark corpus used in the evaluation. 92

List of Figures

2.1 Flow diagram of the SEIR model. Arrows represent infection, latency, recovery, loss of immunity, births, and deaths between compartments. 7

3.1 Class diagram of the EpiMDE metamodel. The root `CompartmentalModel` contains compartments, flows, groups, products, parameters, and external sources/sinks. 16
3.2 EpiMDE diagram of the COVID-19 compartmental model. 20
3.3 EpiMDE diagrams for HIV compartmental models. 21
3.4 The EpiMDE graphical editor (Sirius) showing a simple SIR model. 22

5.1 Overview of the Phase 2 extraction pipeline. 34

6.1 Overview of the Phase 3 gap detection and model completion pipeline. . . . 44
6.2 Three-tier gap filling in Phase 3. 49

7.1 Monte Carlo results for the COVID-19 model. 62
7.2 Monte Carlo results for the HIV model. 63
7.3 Sensitivity tornado plot for the COVID-19 model. 64
7.4 Sensitivity bar chart for the influenza model. 65

8.1 Hierarchical feature tree showing the epidemiological features organised by category. 74
8.2 Feature profile for malaria. 74
8.3 Feature profile for tuberculosis. 75

Abbreviations

- **AI**: Artificial Intelligence
- **API**: Application Programming Interface
- **BM25**: Best Match 25 (probabilistic information retrieval model)
- **CLI**: Command-Line Interface
- **CSV**: Comma-Separated Values
- **DSL**: Domain-Specific Language
- **EMF**: Eclipse Modeling Framework
- **EpiMDE**: Epidemiological Model-Driven Engineering (platform)
- **F1**: F1 score (harmonic mean of precision and recall)
- **FCA**: Formal Concept Analysis
- **FM**: Feature Model
- **GROBID**: GeneRation Of BIbliographic Data (PDF extraction tool)
- **HIV**: Human Immunodeficiency Virus
- **IDE**: Integrated Development Environment
- **JSON**: JavaScript Object Notation
- **LLM**: Large Language Model
- **MDE**: Model-Driven Engineering
- **MDD**: Model-Driven Development
- **MDS**: Model-Driven Software Engineering

- **NLP**: Natural Language Processing
- **OAT**: One-at-a-Time (sensitivity analysis)
- **ODE**: Ordinary Differential Equation
- **PDF**: Portable Document Format
- **P/R/F1**: Precision / Recall / F1 score
- **RAG**: Retrieval-Augmented Generation
- **RQ**: Research Question
- **SEIR**: Susceptible–Exposed–Infectious–Recovered (compartmental model)
- **SEIRD**: Susceptible–Exposed–Infectious–Recovered–Dead
- **SIR**: Susceptible–Infectious–Recovered (compartmental model)
- **SPL**: Software Product Line
- **TB**: Tuberculosis
- **UQ**: Uncertainty Quantification
- **XMI**: XML Metadata Interchange
- **XML**: Extensible Markup Language

Chapter 1

Introduction

1.1 Context and Motivation

Infectious diseases place a recurring burden on human populations. Mathematical models help epidemiologists, public health agencies, and policy makers understand how diseases spread, estimate possible outbreak patterns, and evaluate interventions such as vaccination, physical distancing, and treatment protocols [1, 2]. Compartmental models do this by dividing a population into disease states, or compartments, and defining the rates at which individuals move between them. The resulting system of ordinary differential equations (ODEs) can be used to estimate quantities such as the basic reproduction number, peak prevalence, cumulative incidence, and the effects of public health interventions [3, 4].

The COVID-19 pandemic showed the importance of these models. Research groups and public health agencies produced many compartmental models to study the transmission of SARS-CoV-2, the virus that causes COVID-19, under different assumptions about latent periods, disease severity, interventions, and population structure [5, 6]. However, even when these models used familiar SIR or SEIR structures [7, 8], they were often difficult to compare. Parameters were reported in different units, flow structures were documented with different levels of detail, and assumptions about demography, immunity, and interventions were often described only in prose. This made reproduction and comparison difficult, and limited the reuse of modelling work across studies [9, 10].

The main problem is that publishing a model in a paper is not the same as providing a formal, machine-readable model specification. In this thesis, a formal model specification means a structured EpiMDE `.compmodel` artefact that conforms to an explicit metamodel and can be validated, compared, transformed into equations, and used for simulation. Having this

kind of specification improves reproducibility, verifiability, explainability, and reuse, because the model structure is explicit rather than distributed across prose, equations, diagrams, tables, or implementation code.

ODEs and code can also be formal, but they do not necessarily expose compartments, flows, parameters, and stratification in a reusable and comparable way. A paper may include the necessary information across diagrams, equations, tables, and prose, but reconstructing a simulation-ready model from these pieces still requires expert interpretation and manual encoding [10, 11]. This gap between published model descriptions and structured model specifications is the central problem addressed in this thesis.

1.2 The Case for Model-Driven Engineering

Model-driven engineering (MDE) is useful for this problem because it treats models as structured artefacts rather than informal descriptions or standalone code [12, 13]. In MDE, models follow an explicit metamodel that defines the allowed concepts and relationships in the domain. Once a domain is represented this way, models can be validated, compared, reused, and transformed into code or equations automatically. These are the same properties needed for reproducible compartmental epidemiological modelling.

EpiMDE applies these ideas to epidemiological models [14]. It is the subject of my earlier work, *EpiMDE: A Model-Driven Engineering Platform for Reusable and Reproducible Epidemiological Models*, and provides the modelling foundation used throughout this thesis. EpiMDE defines a formal metamodel for compartments, flows, parameters, and stratification, and provides graphical and textual editors, equation generation, and validation support through the Eclipse Modeling Framework and Sirius [15, 16]. Models created in EpiMDE therefore follow a consistent structure and can be used as formal artefacts for downstream analysis.

However, EpiMDE still assumes that the modeller already knows the structure of the model to encode. It does not solve the earlier step of reading a scientific paper and identifying the compartments, flows, and parameters described in that paper. This thesis focuses on that missing step: moving from document content to a formal EpiMDE model.

1.3 Artificial Intelligence for Model Extraction and Completion

Recent advances in large language models (LLMs) and retrieval-augmented generation (RAG) make it possible to extract structured information from scientific text [17, 18, 19]. This is relevant to epidemiological modelling because papers often contain the information needed to define a model, including compartments, flows, parameters, and initial conditions. The vocabulary is also relatively constrained, and the target output is a formally defined compartmental model.

However, extracting a scientific model is more difficult than generating a natural-language summary. An LLM may produce a model that looks reasonable but still misses important flows, omits parameter values, uses inconsistent names, or adds structure that is not supported by the paper. For this reason, model quality cannot be judged only by whether the output is fluent or plausible. What matters is whether the extracted model matches a reference model and whether it can be simulated without hidden structural errors.

Recent surveys show growing interest in AI-assisted epidemic modelling [20, 21, 22, 23, 24]. However, many existing systems do not evaluate outputs against independently constructed reference models, and few provide traceability links from extracted elements back to the source text. Without traceability, it is difficult to verify which parts of a model are grounded in the paper. Without benchmark evaluation, it is difficult to know where extraction works reliably and where it fails.

1.4 Research Problem and Questions

The central problem addressed in this thesis is:

How can modeling requirements stated in scientific documents be turned systematically into validated, simulation-ready compartmental models, with quantitative evidence on quality and on uncertainty in downstream predictions?

Four research questions organize the investigation:

RQ1: Extraction quality. How accurately does assisted extraction recover compartmental structure (compartments, flows, and parameters) from PDF text relative to hand-authored gold-standard models, and how does performance vary by disease, model complexity, and LLM provider?

RQ2: Completion and residual gaps. When missing parameters and structural elements are addressed through retrieval-augmented completion, which elements are recovered reliably and which systematic categories of gaps remain?

RQ3: Uncertainty and sensitivity. When completion produces models that run reliably, how should uncertainty in epidemic summaries such as peak prevalence and cumulative incidence be described, and which parameters most influence those summaries?

RQ4: Reproducibility of results. How can the pipeline make its runs, configurations, outputs, and evaluation results transparent enough for the main findings to be checked and reproduced?

1.5 Contributions

This thesis makes the following contributions:

1. **An integrated model-driven pipeline.** This thesis presents an end-to-end pipeline that moves from reference-model analysis to LLM-based extraction, retrieval-augmented completion, structural validation, and uncertainty quantification. All phases produce `.compmodel` artefacts that conform to the EpiMDE metamodel and maintain links to the source documents.

2. **A multi-disease benchmark and evaluation protocol.** The evaluation uses a thirty-paper benchmark across ten infectious diseases, with manually constructed gold-standard `.compmodel` files as references. The protocol measures compartment and flow precision and recall, parameter agreement, and overall structural completeness across phases, diseases, and LLM providers.

3. **Feature-level evaluation.** This thesis introduces a complementary evaluation layer that represents each model using seventeen boolean epidemiological features. It compares Phase 2 draft models against reference disease profiles and uses text-grounded recall to focus on features supported by the paper text rather than only by disease-name priors.

4. **Uncertainty quantification and sensitivity analysis.** For structurally valid models, Phase 4 uses Monte Carlo simulation to show how parameter uncertainty affects epidemic outputs. It also applies one-at-a-time sensitivity analysis to identify influential parameters and documents practical simulation issues such as zero estimates, negative rates, and inconsistent initial conditions.

5. **Methodological transparency and reproducibility support.** The thesis documents key design choices, validation rules, scoring formulas, simulation assumptions, and pipeline configurations. This makes the results easier to inspect, reproduce, and extend.[1]

Parts of the work reported in this thesis also form the basis of the submitted paper *From Extraction to Completeness: LLM-Based Reconstruction of Epidemiological Compartmental Models*, which covers the extraction, evaluation, and completion pipeline developed in Phases 1–3.

The main users of this work are computational epidemiology researchers, EpiMDE users, and educators working with formal epidemiological models. The pipeline reduces the manual effort needed to convert a published paper into a structured model and records which parts of the model were recovered automatically, completed, or left for human review.

[1] All pipeline code, benchmark models, configuration files, and evaluation outputs described in this thesis are available at `https://github.com/fatemedbghi/AI-Assisted-Model-Driven-Epidemiology`.

Chapter 2

Background and Related Work

This chapter introduces the background needed to understand the pipeline developed in this thesis. It first reviews compartmental epidemiological models, model-driven engineering, large language models, and retrieval-augmented generation. It then discusses related work on AI-assisted epidemiological modelling, LLM-based extraction from scientific documents, MDE tools for epidemiology, and uncertainty quantification. The chapter ends by positioning this thesis relative to prior work and identifying the gap addressed by the proposed pipeline.

2.1 Compartmental Epidemiological Models

2.1.1 Foundations

Compartmental models are a common mathematical framework for describing how infectious diseases spread within a population [2, 1]. The basic idea is to divide a population of N individuals into mutually exclusive disease states, called *compartments*. Individuals move between these compartments at rates determined by biological and social parameters. These changes over time are represented using a system of ordinary differential equations (ODEs).

This thesis uses the following notation:

- $S(t)$, $E(t)$, $I(t)$, $R(t)$, and $D(t)$ denote the Susceptible, Exposed, Infectious, Recovered, and Dead compartments.
- $N(t)$ denotes the total population at time t.
- β, γ, μ, and σ denote common epidemiological parameters: transmission rate, recovery rate, death or birth rate, and progression rate.

- $\frac{d}{dt}X(t)$ denotes the rate of change of compartment X over time.
- Subscripts, such as I_a or I_s, denote stratified sub-populations.

2.1.2 The SIR Model

The SIR model, introduced by Kermack and McKendrick and later systematized by Hamer [8], divides a population into Susceptible (S), Infectious (I), and Recovered (R) compartments. Its equations are:

$$\frac{dS}{dt} = -\frac{\beta\, S\, I}{N} \tag{2.1}$$

$$\frac{dI}{dt} = \frac{\beta\, S\, I}{N} - \gamma I \tag{2.2}$$

$$\frac{dR}{dt} = \gamma I \tag{2.3}$$

The term $\beta SI/N$ represents infection. It depends on both the susceptible population and the infectious population, so it is treated as a contact flow. The term γI represents recovery. It depends only on the infectious population, so it is treated as a rate flow. The parameter γ also defines the average infectious period, which is $1/\gamma$ days.

The SIR model is useful for diseases where recovered individuals gain lasting immunity. It also introduces the basic reproduction number:

$$R_0 = \frac{\beta}{\gamma} \tag{2.4}$$

When $R_0 > 1$, one infectious individual causes more than one new infection on average, so the outbreak grows. When $R_0 < 1$, the outbreak declines [3].

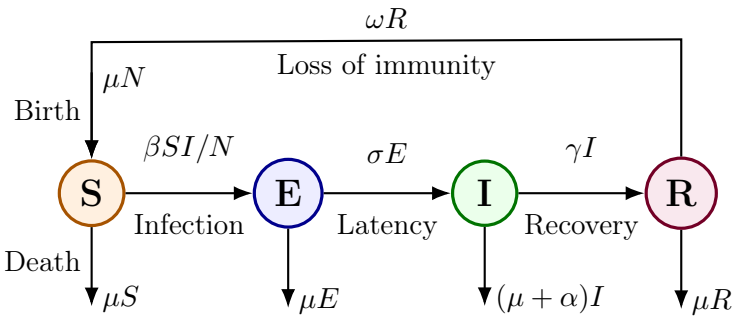


Figure 2.1: Flow diagram of the SEIR model. Arrows represent infection, latency, recovery, loss of immunity, births, and deaths between compartments.

2.1.3 The SEIR Model and Extensions

The SEIR model extends the SIR model by adding an Exposed (E) compartment. This compartment represents the latent period between infection and becoming infectious [7]. The equations are:

$$\frac{dS}{dt} = -\frac{\beta\, S\, I}{N} \tag{2.5}$$

$$\frac{dE}{dt} = \frac{\beta\, S\, I}{N} - \sigma E \tag{2.6}$$

$$\frac{dI}{dt} = \sigma E - \gamma I \tag{2.7}$$

$$\frac{dR}{dt} = \gamma I \tag{2.8}$$

where σ is the progression rate from exposed to infectious. The average latent period is therefore $1/\sigma$ days.

Real diseases often require more detailed structures than SIR or SEIR. Published models may add compartments for hospitalization, death, vaccination, quarantine, asymptomatic infection, or treatment. They may also stratify the population by age, sex, geography, risk behaviour, or vaccination status [25, 26]. For example, an SEIR model with three age groups and two sex groups contains $4 \times 3 \times 2 = 24$ state variables.

Some diseases require additional modelling structures. Vector-borne diseases such as malaria, dengue, and Zika use both human and vector populations, linked through contact flows such as mosquito biting [27, 28, 29]. Sexually transmitted infections such as HIV often require stratification by risk behaviour and may include treatment compartments for antiretroviral therapy [30].

2.1.4 Flow Types

Three flow types appear throughout this thesis and are central to the EpiMDE metamodel design described in Chapter 3:

- **Contact flows.** These represent transmission. They depend on both the source compartment and a contact compartment. The standard example is $\beta S(t)I(t)/N(t)$, where infection depends on contact between susceptible and infectious individuals [31].

- **Rate flows.** These represent transitions such as progression, recovery, or mortality. They depend only on the source compartment. For example, $\gamma I(t)$ represents recovery from the infectious compartment.

- **External sources and sinks.** These represent movement into or out of the modelled population, such as births, natural deaths, immigration, or imported cases. They are treated separately from disease-state transitions.

2.1.5 Challenges of Transcribing Published Models

A major challenge in using published compartmental models is that the full model specification is rarely presented in one place. Compartments may appear in the methods section, parameters in a table, equations in a supplement, and initial conditions in a short sentence. Important details, such as the exact flow structure or stratification scheme, often require expert interpretation [10, 6].

This gap between the paper as a written description and the model as a computational artefact is the main motivation for this thesis. When this gap is large, reproduction becomes difficult, comparison across studies becomes unreliable, and model reuse becomes slower [9, 32].

2.2 Model-Driven Engineering

2.2.1 Core Principles

Model-driven engineering (MDE) is a software engineering approach that treats models as primary development artefacts [12, 13]. Instead of making program code the main representation, MDE uses models that follow an explicit *metamodel*. A metamodel defines the concepts, attributes, and relationships that are valid in a domain. A model is valid when it follows these rules.

This is useful for epidemiological modelling because it supports:

- **Consistency.** Models that follow the same metamodel use the same vocabulary and structure, which makes comparison easier.

- **Automation.** Equations, simulation code, documentation, or other representations can be generated from the model.

- **Traceability.** Model elements can be linked to their sources, making review and verification easier.
- **Reuse.** Model components can be stored and reused across projects without rebuilding the same structure manually.

2.2.2 Metamodels and Domain-Specific Languages

A metamodel defines a domain-specific language (DSL) for a particular modelling domain. In the Eclipse Modeling Framework (EMF) [15], metamodels are defined as Ecore schemas made of classes, attributes, references, and containment relationships. Model instances are stored as XML Metadata Interchange (XMI) files. Sirius extends EMF with graphical editing support, allowing users to work with models through diagrams instead of editing raw XML directly [16].

Metamodel-based DSLs have been used in several domains, including healthcare digital twins [33] and AI system engineering [34]. Recent work also shows increasing interest in combining MDE with machine learning and AI components [35, 36].

2.2.3 MDE in Epidemiology

EpiMDE applies MDE principles to epidemiological modelling [14]. The motivation is that epidemiological models are often difficult to compare when they are created with different tools, formats, and assumptions. A shared metamodel provides a common structure for representing compartments, flows, parameters, and stratification.

With this structure, models can be validated, transformed, compared, and simulated more systematically. Chapter 3 describes the EpiMDE metamodel and platform in detail.

2.3 Large Language Models and Retrieval-Augmented Generation

In this thesis, large language models are used to extract structured model information from scientific papers. Within the proposed pipeline, they help identify compartments, flows, parameters, and supporting evidence from text, equations, and tables, and convert these elements into outputs that can be mapped to EpiMDE model artefacts. Their role is not to

replace expert judgement or model validation, but to support a controlled extraction and completion process guided by prompts, schemas, traceability links, and quality checks.

2.3.1 Transformer Architecture

Modern large language models are based on the Transformer architecture introduced by Vaswani et al. [37]. Transformers use a self-attention mechanism to process text. This allows the model to consider relationships between words across a long passage, rather than only reading words in sequence. This architecture is one reason LLMs are effective at reading, summarising, extracting information, and generating structured outputs.

Models such as GPT-4 [17], Gemini [18], and Claude [38] can perform tasks such as reading comprehension, structured extraction, and code generation. When prompted carefully, they can return outputs in formats such as JSON, XML, or domain-specific structures.

2.3.2 LLMs for Structured Extraction

LLMs are useful for extracting structured information from scientific papers because they can identify entities, relationships, and descriptions from natural language. In this thesis, the relevant entities include compartments, flows, parameters, and initial conditions. Prior work has explored LLM-based extraction in biological, medical, and epidemiological contexts [20, 39, 21, 40].

The main challenge is faithfulness. The extracted model must reflect what the paper actually states, not what the LLM assumes is likely based on general knowledge. This is especially important for parameters and flows, because a plausible but incorrect value or transition can produce incorrect simulations.

To address this, the pipeline compares extracted models against hand-authored reference models and links extracted elements back to source text spans. This makes it possible to check whether each extracted compartment, flow, or parameter is supported by the paper.

2.3.3 Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) combines language generation with document retrieval [19]. Instead of relying only on the LLM's internal knowledge, the system first searches a document collection for relevant passages. These passages are then provided to the LLM as additional context.

Retrieval can be done using keyword-based methods, dense vector similarity, or a combination of both. For example, Sentence-BERT can represent sentences as vectors and retrieve passages with similar meaning even when the wording is different [41].

In this thesis, RAG is used in Phase 3 to help complete missing model elements. When Phase 2 does not recover a compartment, flow, or parameter, Phase 3 searches the indexed paper database for relevant evidence and provides that evidence to the LLM. This helps the completion step remain grounded in source material and keeps filled elements traceable to specific passages.

2.4 Related Work

2.4.1 AI-Assisted Epidemiological Modeling

Artificial intelligence is increasingly used in infectious disease modeling. Kraemer et al. [20] survey AI applications across the epidemic modeling pipeline, including outbreak detection, genomic surveillance, transmission modeling, and intervention analysis. They show that AI can support mechanistic modeling, but also note that AI outputs are not yet well integrated into formal modeling workflows.

Ye et al. [23] review work that combines AI with mechanistic epidemiological models. They find that most studies focus on parameter estimation, meaning fitting model parameters to observed data. This is different from the problem addressed in this thesis, which is extracting model structure from published papers.

Kwok et al. [21] show that LLMs can help public health practitioners design transmission models through conversation. However, their workflow produces Python code rather than a structured model that follows a formal metamodel, and the resulting models are not evaluated against reference models. Other work, such as Du et al. [42], focuses on infectious disease forecasting rather than model structure extraction.

Matthay et al. [22] discuss opportunities for AI in causal epidemiological research and emphasize the need for rigorous evaluation. Chopra et al. [43] study differentiable agent-based epidemiology, which is a different modeling approach from the compartmental ODE models used in this thesis.

2.4.2 LLM-Based Extraction from Scientific Documents

LLMs have also been used to extract structured information from scientific and medical text. Wibaek et al. [40] use LLMs to support automated machine learning in epidemiological research, but their target is statistical model specification rather than dynamic compartmental model structure.

EpidemIQs [44] uses LLM agents for epidemic modeling and analysis from natural language prompts, while EpiLLM [45] applies LLMs to epidemic forecasting from time-series data. These systems address related tasks, but they do not focus on extracting formal compartmental models from existing scientific papers.

This thesis builds on my earlier work on generating EpiMDE-compatible `.compmodel` files from scientific papers [24]. That study used Gemini 2.5 Pro with text, figures, and tables to reconstruct three disease models, and evaluated the outputs using element-level precision, recall, and F1. The present thesis extends the scope to thirty papers across ten diseases, adds retrieval-augmented completion and uncertainty analysis, and introduces feature-level evaluation.

2.4.3 MDE Tools for Epidemiology and Related Domains

Several tools support structured epidemiological modeling. Kendrick [46] provides a domain-specific language for stochastic epidemiological models. EpiModel [47] supports infectious disease modeling over contact networks in R. STEM [48] is an Eclipse-based disease modeling tool with a graphical interface. FlepiMoP [49] provides a flexible modeling pipeline developed during the COVID-19 pandemic.

These tools support model construction and simulation, but they do not provide the same combination of formal EMF/Sirius-based metamodeling, equation generation from model instances, and AI-assisted extraction from scientific papers. EpiMDE [14] is the closest platform in this respect, and this thesis extends it with an automated paper-to-model pipeline.

MDE methods have also been used in related domains such as healthcare digital twins [33] and AI system engineering [34, 50]. These examples show that model-driven methods can support structured computational modeling beyond traditional software systems.

2.4.4 Uncertainty Quantification in Epidemic Modeling

Uncertainty quantification is a standard part of epidemic modeling [51]. Monte Carlo simulation samples parameter values from probability distributions and uses the resulting

set of simulations to describe uncertainty in model outputs, such as peak prevalence and final epidemic size. One-at-a-time sensitivity analysis [52] provides a simple way to identify influential parameters, while global methods such as Sobol indices [53] provide a more complete but more expensive analysis.

Most uncertainty methods assume that the model structure is already correct. This assumption is difficult in the setting of this thesis, because extracted models may still contain missing flows, incomplete parameters, or unsupported elements. For this reason, Phase 4 applies uncertainty quantification only to structurally valid models and reports problematic outputs separately.

2.4.5 Reproducibility in Computational Epidemiology

Reproducibility remains a major issue in computational epidemiology [10, 6, 9]. Jalali et al. [10] show that many simulation studies lack sufficient documentation of model structure, parameters, and software environments. Watanabe et al. [32] discuss SBML as a possible structured exchange format for disease models, while also noting practical barriers to standardized model sharing.

This thesis supports reproducibility through structured model artifacts, explicit traceability links, versioned configurations, and documentation of methodological choices.

2.4.6 Positioning This Thesis

The related work shows a clear gap. Existing systems support AI-assisted modeling, structured epidemiological tools, or uncertainty analysis, but they do not combine these pieces into a single documented pipeline for extracting formal compartmental models from scientific papers. This thesis addresses that gap by combining the EpiMDE metamodel with LLM-based extraction, retrieval-augmented completion, multi-disease benchmark evaluation, uncertainty quantification, and feature-level evaluation. Compared with prior work, the main additions are span-level traceability to source passages, evaluation across thirty papers and three LLM providers, and a feature-vector method for checking whether extracted models preserve disease-level epidemiological traits.

Chapter 3

The EpiMDE Platform

EpiMDE is a model-driven engineering platform for building, visualising, and analysing compartmental epidemiological models. Instead of representing models as scattered equations, code, or spreadsheets, EpiMDE stores them as structured `.compmodel` artefacts that follow a formal metamodel. This makes the models easier to validate, compare, transform, and reuse.

This chapter presents the EpiMDE platform as the formal modelling foundation for the pipeline developed in this thesis. It describes the platform's design goals, metamodel, example disease models, Eclipse-based modelling environment, equation generation, and simulation support. The chapter also identifies the main limitation that motivates the AI-assisted pipeline: EpiMDE provides a formal modelling environment, but it does not automatically extract model structure from scientific papers.

3.1 Design Goals

The design of EpiMDE was guided by three main requirements from epidemiologists and public health modellers:

1. **Domain familiarity.** The platform should use concepts that epidemiologists already know, such as compartments, flows, stratification, and parameters. Users should not need to understand MDE or Eclipse internals to build valid models.

2. **Automation.** The platform should reduce the manual work required to derive equations from model structure. This is especially important for stratified models, where the number of state variables and equations can grow quickly.

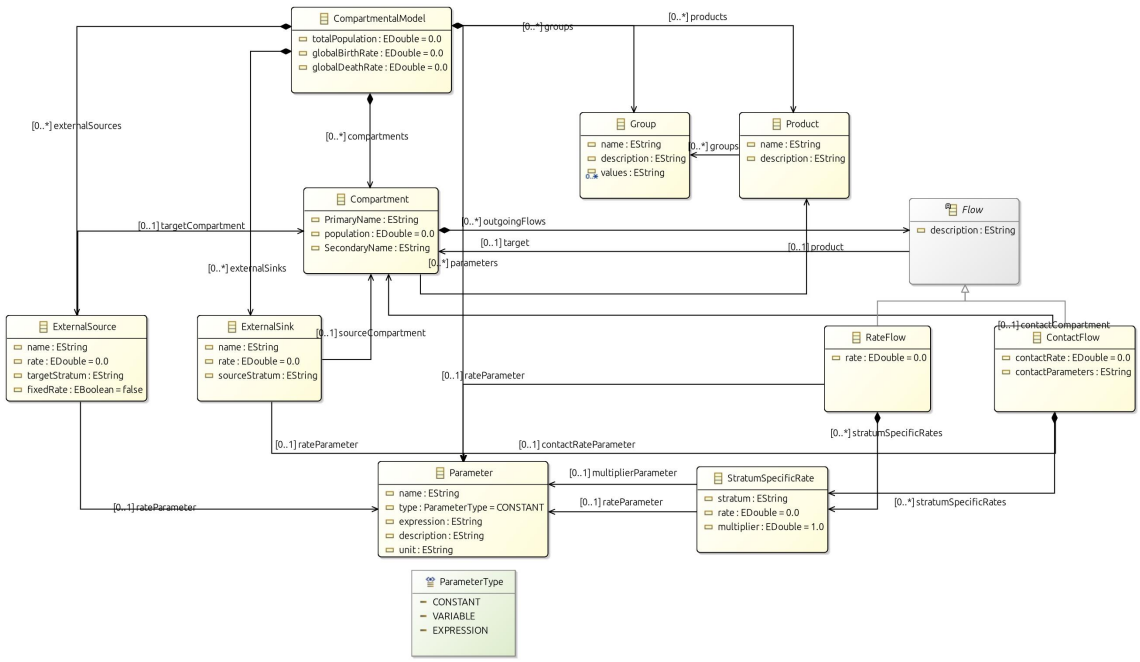


Figure 3.1: Class diagram of the EpiMDE metamodel. The root `CompartmentalModel` contains compartments, flows, groups, products, parameters, and external sources/sinks.

3. **Reproducibility and comparability.** Models stored as structured metamodel artefacts can be version-controlled, compared, and reused more easily. This helps address reproducibility problems in epidemiological modelling [10, 6].

For this thesis, EpiMDE also needs to be machine-readable. The `.compmodel` XML format is used as the target format for the AI-assisted pipeline. It serves both as the output format for extracted draft models and as the reference format for the hand-authored gold-standard models.

3.2 The EpiMDE Metamodel

3.2.1 Overview

The EpiMDE metamodel is defined as an Ecore schema in the Eclipse Modeling Framework [15]. The root element is `CompartmentalModel`, which contains the full model, including compartments, flows, groups, products, parameters, external sources, and external sinks. Figure 3.1 shows the class diagram, and Table 3.1 summarises the main metamodel elements.

Table 3.1: Core EpiMDE metamodel elements and their epidemiological roles.

Element	Type	Epidemiological Role
`CompartmentalModel`	Root	Container for the full model, including population size, global birth rate, and global death rate.
`Compartment`	Class	Disease state such as Susceptible, Exposed, Infectious, Recovered, or Death.
`RateFlow`	Flow	Transition proportional to the source compartment, such as recovery or disease progression.
`ContactFlow`	Flow	Transmission flow depending on both the source compartment and a contact compartment.
`ExternalSource`	Source	Population inflow, such as births into a susceptible compartment.
`ExternalSink`	Sink	Population outflow, such as natural mortality.
`Group`	Stratification	Stratification dimension with named values, such as age groups.
`Product`	Stratification	Cartesian product of referenced groups, defining the complete set of strata.
`StratumSpecificRate`	Rate detail	Per-stratum override of a flow rate using either an absolute rate or a multiplier.
`Parameter`	Parameter	Named model quantity with a type, optional unit, description, or expression.

3.2.2 Compartments and Flows

A `Compartment` represents a disease state, such as Susceptible, Exposed, Infectious, Recovered, or Death. Each compartment has a primary name, an initial population, and optionally a reference to a `Product` when the compartment is stratified.

EpiMDE supports two main flow types:

RateFlow. A rate flow represents a transition that depends only on the source compartment. Its ODE term has the form $rate \times C_{src}$, where C_{src} is the source compartment population. Rate flows are used for transitions such as progression, recovery, and disease mortality.

ContactFlow. A contact flow represents transmission. It depends on the source compartment and a contact compartment, usually an infectious compartment. A typical term is $contactRate \times S \times I/N$. Contact flows encode mass-action transmission in SIR and SEIR-type models. For vector-borne diseases, the contact compartment may be an infectious vector compartment.

3.2.3 Population Stratification

EpiMDE represents stratification using `Group`, `Product`, and the `product` reference on `Compartment`. A `Group` defines one stratification dimension, such as age or sex. A `Product` combines one or more groups into the full set of strata.

For example, an age group with three values and a sex group with two values create six strata. A compartment linked to this product is expanded into one state variable per stratum when equations are generated. This keeps population structure separate from the disease-flow structure.

`StratumSpecificRate` elements allow rates to vary by stratum. A flow can therefore have a base rate, with specific strata using an adjusted value or multiplier.

3.2.4 External Sources and Sinks

External sources and sinks represent movement into or out of the modelled population. Examples include births, natural deaths, immigration, and emigration. They are stored at the model root rather than as ordinary compartment-to-compartment flows, because they describe population entry or exit rather than disease-state transitions.

3.2.5 The Parameter System

EpiMDE separates model structure from numeric values through named parameters. It supports three parameter types:

- **CONSTANT.** A fixed numeric value, such as a recovery rate.
- **VARIABLE.** A value supplied later, such as an intervention level or time-varying quantity.
- **EXPRESSION.** A symbolic expression derived from other parameters, such as $\lambda = \beta \cdot I/N$.

Parameters are defined at the model root and referenced by flows. This allows the same model structure to support both numeric and symbolic parameter definitions.

3.3 Disease Model Examples

3.3.1 Overview of Bundled Models

The EpiMDE repository includes 30 hand-authored reference models across 10 diseases, with three models per disease. Each model is encoded as a `.compmodel` file. The examples cover simple SIR and SEIR structures as well as more detailed models with stratification, clinical progression, hospitalization, intervention policies, and demographic structure.

This section highlights two representative examples: a COVID-19 model and an HIV model.

3.3.2 The COVID-19 Model

The COVID-19 example reproduces the structure of the Tuite et al. Ontario transmission model [54]. It extends the basic SEIR structure with compartments for asymptomatic infection, mild disease, severe disease, hospitalization, ICU admission, recovery, and disease death. It also includes intervention parameters related to physical distancing.

Figure 3.2 shows the EpiMDE diagram for this model. Contact flows represent infection, while rate flows represent clinical progression between disease stages.

3.3.3 The HIV Model

The HIV example illustrates stratification. HIV transmission risk differs across population groups, so the model includes stratification by gender and risk group. It also includes treatment through antiretroviral therapy (ART), which affects infectiousness and disease progression [30].

Figure 3.3 shows the full HIV model and a variant focused on women with empowerment-status stratification. These examples show how EpiMDE can represent both disease progression and population heterogeneity.

3.4 The EpiMDE IDE

3.4.1 Architecture

The EpiMDE IDE is built on Eclipse Modeling Tools, the Eclipse Modeling Framework (EMF) [15], and Sirius [16]. Figure 3.4 shows the graphical editor with a simple SIR model.

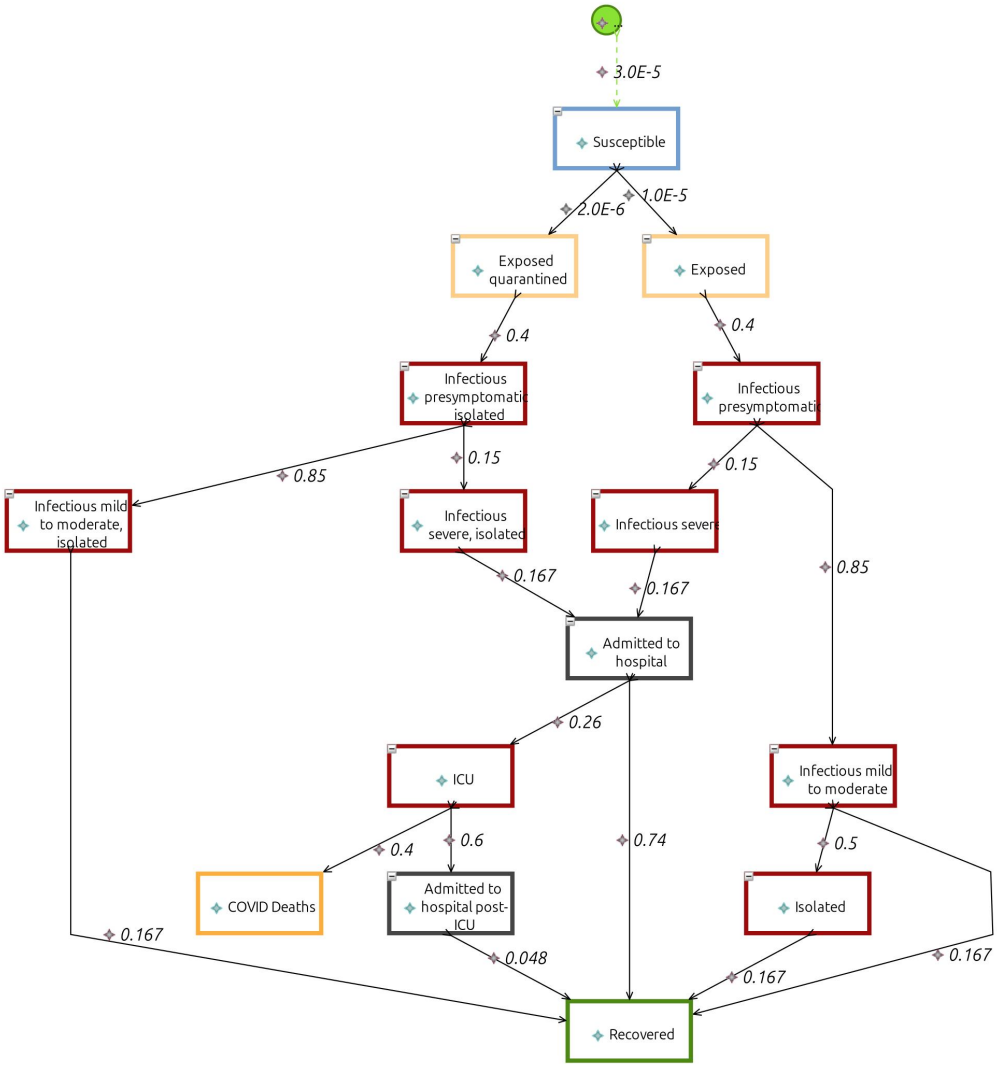
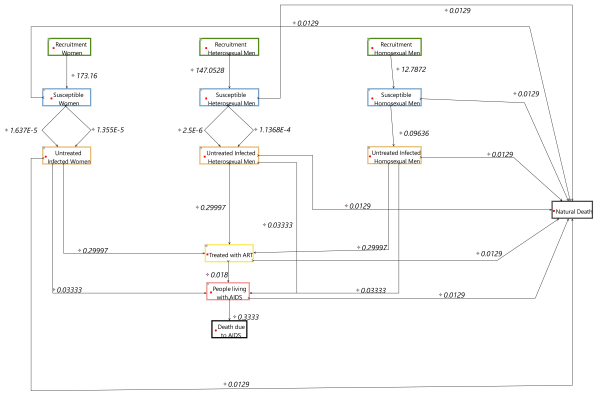
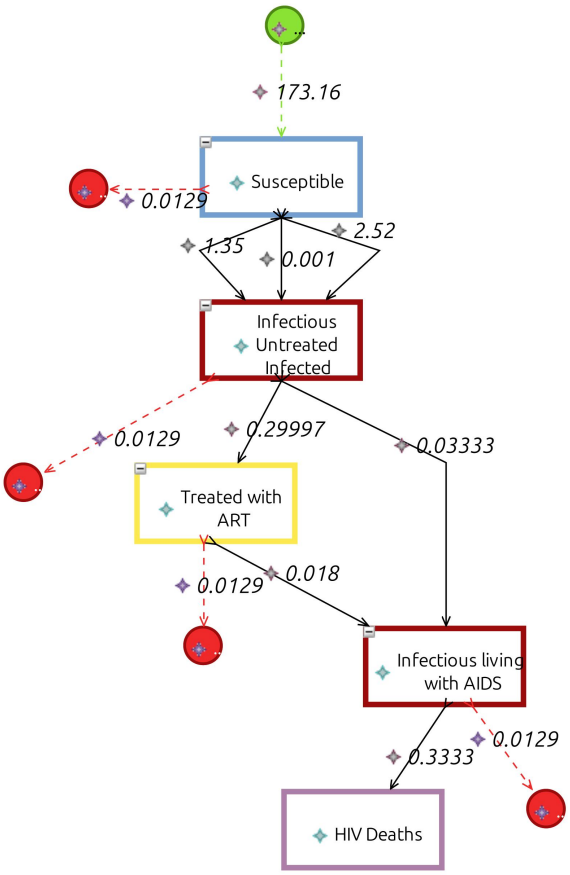


Figure 3.2: EpiMDE diagram of the COVID-19 compartmental model.



(a) Full HIV model with male and female strata.



(b) HIV model variant focused on women with stratification by empowerment status, drawn from the same source paper as the full model [30].

Figure 3.3: EpiMDE diagrams for HIV compartmental models.

The IDE includes four Eclipse projects:

- **CompartmentalModel**: the core EMF project containing the Ecore metamodel, generated Java API, and example `.compmodel` files.
- **CompartmentalModel.edit**: EMF item providers for displaying and editing model elements.
- **CompartmentalModel.editor**: the standard EMF tree editor plugin.
- **CompartmentalModel.design**: the Sirius `.odesign` file defining the graphical notation, palette, and layout rules.

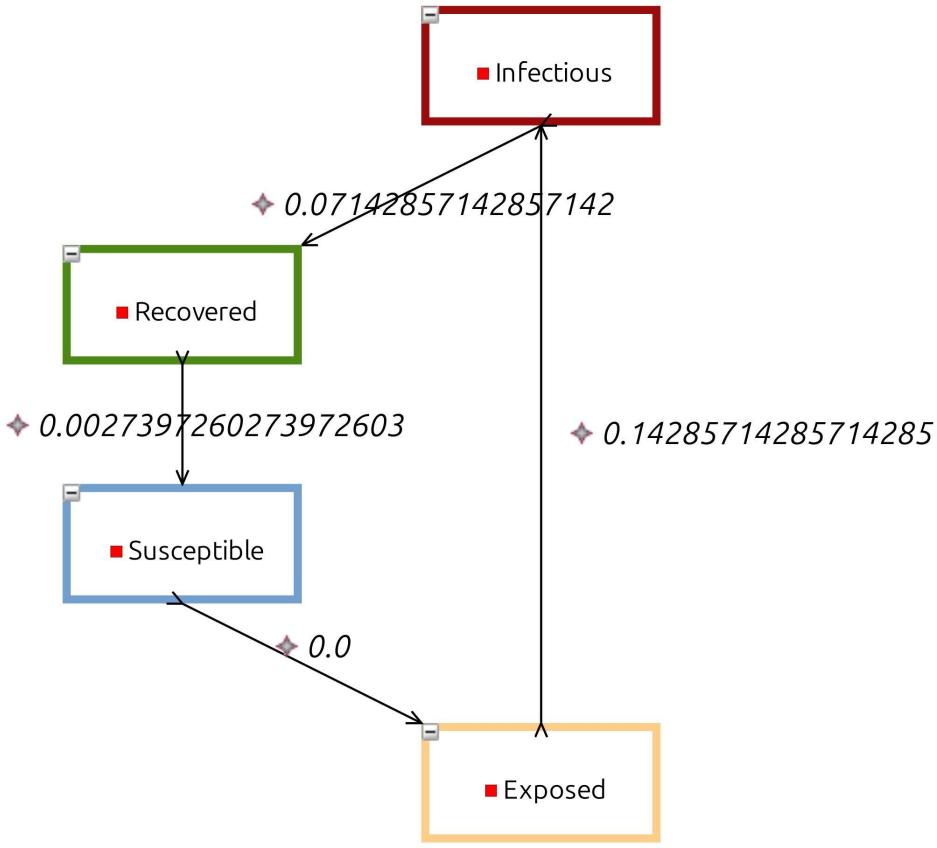


Figure 3.4: The EpiMDE graphical editor (Sirius) showing a simple SIR model.

3.4.2 Graphical and Tree Editors

EpiMDE provides two complementary editing views:

Sirius diagram editor. The graphical editor shows compartments as nodes and flows as directed edges. It is useful for designing and checking the overall model structure.

EMF tree editor. The tree editor shows the full model hierarchy. It is useful for editing detailed attributes, parameter values, stratum-specific rates, and validation issues.

3.5 Equation Generation

3.5.1 CompartmentalEquationGenerator

The `CompartmentalEquationGenerator` reads a `.compmodel` file and generates the corresponding ODE equations. For non-stratified models, this produces one equation per compartment. For stratified models, the generator expands each stratified compartment into one equation per stratum.

For example, a simple COVID-19 model may produce equations such as:

$$\frac{dS}{dt} = -\beta \cdot S \cdot I_{total}/N - \mu \cdot S + \lambda \tag{3.1}$$

$$\frac{dE}{dt} = \beta \cdot S \cdot I_{total}/N - \sigma \cdot E - \mu \cdot E \tag{3.2}$$

$$\frac{dI}{dt} = \sigma \cdot E - \gamma \cdot I - \delta \cdot I - \mu \cdot I \tag{3.3}$$

where λ is the birth rate, μ is the natural mortality rate, δ is disease-induced mortality, and I_{total} aggregates infectious sub-compartments.

3.5.2 Python Simulation

EpiMDE includes a basic Python simulation script that reads generated equations, asks the user for initial conditions and a time horizon, and numerically integrates the ODE system. This is useful for simple models, but it has limitations. It handles only relatively simple equation forms, requires interactive input, and does not use adaptive step-size control.

For this reason, the uncertainty quantification phase in Chapter 7 uses a separate simulation implementation designed for repeated Monte Carlo runs.

3.6 The Role of EpiMDE in the AI-Assisted Pipeline

EpiMDE plays three main roles in the AI-assisted pipeline described in Chapters 4–8:

1. **Reference format.** The hand-authored gold-standard models are encoded as `.compmodel` files, ensuring that evaluation uses formally structured references.

2. **Output format.** Phase 2 synthesizes extracted model elements into `.compmodel` XML. The metamodel therefore constrains the generated output and makes invalid structures detectable.

3. **Evaluation standard.** Phases 2 and 3 compare extracted and completed `.compmodel` files against the reference `.compmodel` files using structural matching.

The main limitation is that EpiMDE does not create a model from a paper by itself. A modeller still has to read the source paper, identify compartments, flows, and parameters, and encode them manually. The AI-assisted pipeline developed in this thesis targets this missing step by generating a draft model automatically, so that the modeller can review and correct it rather than starting from scratch.

Chapter 4

Phase 1: Baseline Analysis

Phase 1 establishes the reference foundation for the pipeline. It analyzes the hand-authored gold-standard `.compmodel` files, describes their structural properties, compares them with the modelling elements promised in the source papers, documents parameter uncertainty, and identifies sensitive parameters.

The outputs of Phase 1 are used in two ways. First, the gold-standard models provide the reference for evaluating Phases 2 and 3. Second, the shared artefacts produced in this phase help guide LLM prompting in Phase 2.

4.1 Motivation and Goals

Automated extraction requires a reliable reference. If the gold-standard models are incomplete or inconsistent, the evaluation would measure problems in the reference rather than problems in the extraction. For this reason, Phase 1 first analyzes the reference models before they are used for evaluation.

Phase 1 has four goals:

1. **Structural characterization.** Record the compartments, flow types, named parameters, and stratification structures present in each gold-standard model.
2. **Paper promise analysis.** Identify what each source paper claims to model and compare this with what the corresponding gold-standard model encodes.
3. **Parameter uncertainty documentation.** Record parameter values, available literature ranges, confidence labels, and parameters that are missing or underspecified.

4. **Sensitivity screening.** Identify which parameters most strongly influence epidemic outputs, providing context for Phase 4 uncertainty quantification.

4.2 Gold-Standard Corpus

4.2.1 Corpus Construction

The gold-standard corpus contains thirty hand-authored `.compmodel` files, with three papers for each of ten diseases. Table 4.1 lists the diseases and their primary reference papers.

Table 4.1: Ten-disease benchmark corpus used in the evaluation.

Disease	**Primary Reference**	**Disease Class**
COVID-19	Tuite et al. (2020) [54]	Respiratory
HIV	Espitia et al. (2022) [30]	Sexually transmitted
Malaria	Akowe et al. (2025) [29]	Vector-borne
Dengue	Li et al. (2023) [27]	Vector-borne
Ebola	Abbas et al. (2024) [55]	Hemorrhagic fever
Measles	Pokharel et al. (2024) [56]	Vaccine-preventable
Tuberculosis	Ginting et al. (2024) [57]	Respiratory / chronic
Zika	Lee et al. (2016) [28]	Vector-borne
Cholera	Fung (2014) [58]	Waterborne
Influenza	Corbella et al. (2017) [59]	Respiratory

The diseases were selected to cover different transmission mechanisms and model structures. Respiratory diseases such as COVID-19, tuberculosis, and influenza are often described with structured parameters. Vector-borne diseases such as malaria, dengue, and Zika require human and vector populations. HIV requires behavioural stratification, while cholera requires environmental transmission. This variety allows the pipeline to be evaluated across different modelling styles.

4.2.2 Benchmark Layout

Each benchmark item consists of a source paper and a paired `.compmodel` reference model. Each reference model was created by reading the paper, identifying compartments and flows, recording parameter values, and encoding stratification where needed.

4.3 Structural Analysis

4.3.1 Methodology

Each `.compmodel` file is parsed to extract the following structural features:

- Number of compartments, including disease-state categories such as Susceptible, Exposed, Infectious, Recovered, and Dead
- Number of rate flows and contact flows
- Number of external sources and sinks
- Presence and size of stratification structures
- Number of named parameters, classified by type
- Boolean flags for stratification, birth, natural death, and disease death

4.3.2 Results

Table 4.2 summarizes the structural analysis results across the ten primary gold-standard models.

The structural analysis shows clear differences across diseases. Vector-borne diseases have more contact flows because they model interactions between human and vector compartments. HIV has the most stratification because it separates the population by sex and risk group. Model size also varies, from smaller models such as cholera and Ebola to larger models such as COVID-19. These differences affect the difficulty of Phase 2 extraction.

Table 4.2: Structural characteristics of the ten primary gold-standard models.

Disease	Comp.	R-flows	C-flows	Params	Strata	Birth/death
COVID-19	9	7	1	11	1	Yes
HIV	8	6	2	14	4	Yes
Malaria	8	6	4	10	1	Yes
Dengue	7	5	3	9	1	Yes
Ebola	6	5	1	8	1	No
Measles	5	4	1	7	3	Yes
Tuberculosis	7	6	1	10	1	Yes
Zika	8	5	4	11	1	Yes
Cholera	5	4	1	7	1	Yes
Influenza	6	4	1	8	3	Yes

4.4 Gap Analysis: Paper Promises vs. Model Content

4.4.1 Defining Paper Promises

A *paper promise* is a modelling element that the source paper claims to include. This may be a named compartment, a described flow, a parameter value, or a stratification structure. Gap analysis checks whether each promised element is present in the gold-standard model.

This analysis identifies cases where the reference model may not include everything stated in the paper. It also helps characterize the types of paper-to-model gaps that motivate Phase 3 completion.

4.4.2 Promise Extraction Methodology

Two promise extraction methods are used:

- **Pattern-based.** Regular expressions match common epidemiological phrases such as "compartments include", "SEIR model", "stratified by", and "parameters are listed in". This is the default method because it does not require API access.
- **LLM-based.** An LLM reads the relevant paper sections and lists the modelling elements that appear to be explicitly described. This method is more flexible, but it may also produce unsupported promises.

Promise extraction is imperfect. Pattern-based extraction can miss subtle wording, while LLM-based extraction can over-infer. To check reliability, both methods were compared against a manually annotated subset of five papers. Pattern-based extraction reached 78% agreement for compartments and 61% agreement for parameters.

4.4.3 Gap Analysis Results

Across the ten primary papers, three common patterns appear:

1. **Implicit parameters.** Some papers describe a flow without providing a numeric rate in the main text. The value may be in supplementary material, another publication, or estimated from data.

2. **Aggregated compartments.** Some papers describe multiple disease states, but the gold-standard model merges them into a simpler compartment.

3. **Implicit birth/death.** Some papers assume a static population, while the gold-standard model includes birth or death flows for demographic balance.

These patterns show why evaluation needs to distinguish between missing elements, unsupported extras, and differences caused by modelling interpretation.

4.5 Parameter Uncertainty Documentation

4.5.1 Methodology

For each named parameter in every gold-standard model, Phase 1 records:

- the parameter name, type, and value;
- a `literature_range`, when available;
- a `confidence` label: `High`, `Medium`, `Low`, or `Unknown`.

The `Unknown` label is used when no reliable range or confidence information is available. Fully populating these labels would require systematic links to external parameter databases or meta-analyses, which is outside the scope of this thesis.

4.5.2 Key Findings

Across the benchmark, some parameters are relatively well characterized, such as recovery rate γ and natural mortality μ. Other parameters are more uncertain, especially disease-specific mortality rates, contact rates, and behavioural parameters. Many of these values are estimated from specific outbreak data and are not directly supported by broader literature ranges.

4.6 Sensitivity Analysis

4.6.1 Methodology

Sensitivity analysis measures how much a model output changes when a parameter changes. Parameters that strongly affect the output are important for later uncertainty analysis.

Four complementary methods are used, ordered roughly from fastest to most thorough:

- **One-at-a-time (OAT) grid**: One parameter is varied while all others remain fixed. This is simple and interpretable, but it does not capture interactions between parameters.
- **Morris screening** [52]: Parameters are varied one at a time from multiple random starting points. This shows both how influential a parameter is and whether its effect depends on other parameter values.
- **Random sampling**: All parameters are varied simultaneously within a fixed range. The spread in outputs gives a quick estimate of overall uncertainty.

4.6.2 Results

Sensitivity patterns are broadly consistent across diseases. The transmission rate β and progression rate σ have the strongest effect on peak prevalence. The recovery rate γ has the strongest effect on final epidemic size, because faster recovery reduces the total number of infections.

Demographic parameters such as birth rate and natural mortality have little effect in short-term simulations because demographic change is slower than epidemic spread. They become more important in longer-term steady-state analyses.

4.7 Shared Artifacts for Downstream Phases

Phase 1 produces artefacts that are reused by later phases:

- **Metamodel description file.** This machine-readable file describes the EpiMDE modelling language. Phase 2 uses it to guide LLM prompts and keep generated outputs consistent with the metamodel.
- **Structural summaries.** These summaries record the number of compartments, flows, and parameters in each reference model. Phase 2 uses them to identify extracted drafts that appear too small or unusually complex.
- **Pattern library.** This library records common structures such as SIR, SEIR, and vector-borne disease patterns. Phase 3 uses it when completing partially extracted models.
- **Disease taxonomies.** Diseases are grouped by transmission and structural characteristics. Phase 3 uses these groupings to retrieve more relevant information during completion.

4.8 Chapter Summary

Phase 1 establishes the reference foundation for the pipeline. It builds the ten-disease, thirty-paper benchmark using hand-authored `.compmodel` files and analyzes model structure, paper promises, parameter uncertainty, and parameter sensitivity.

These outputs directly support the later phases. The gold-standard models provide the evaluation reference for Phases 2 and 3. The metamodel artefact guides LLM extraction in Phase 2. The sensitivity results support distribution assignment in Phase 4. Overall, Phase 1 ensures that the pipeline is built on a documented reference corpus rather than an assumed ground truth.

The next chapter describes Phase 2, which uses the reference corpus and metamodel context to extract compartmental models from scientific PDFs.

Chapter 5

Phase 2: Model Extraction from Scientific Literature

Phase 2 extracts draft compartmental models from peer-reviewed epidemiological papers. It takes a PDF as input and produces a populated `.compmodel` XML file that conforms to the EpiMDE metamodel. It also produces structured JSON artefacts that record the extracted entities, source passages, quality checks, and evaluation results.

This chapter describes the motivation for Phase 2, the seven-step extraction pipeline, the benchmark and evaluation setup, the LLM provider configurations, the extraction results, the main error types, and the traceability mechanism.

5.1 Motivation and Design Goals

Epidemiological papers describe models through a mixture of prose, equations, diagrams, and parameter tables. Manually converting these descriptions into machine-readable models is slow and error-prone. Phase 2 automates this step by using LLM-based extraction guided by the EpiMDE metamodel.

5.1.1 The Faithfulness Principle

The main design principle of Phase 2 is *faithfulness*. The pipeline should extract only the model elements clearly described in the source paper. It should not complete the model using outside knowledge or assumptions.

This matters because an extracted model can look plausible while still adding unsupported compartments, flows, or parameters. For example, if a paper describes only a four-compartment SEIR model, Phase 2 should not add extra compartments unless the paper explicitly includes them. Unsupported additions are treated as hallucinations and reduce precision.

This principle is enforced in three ways. The prompt instructs the LLM to extract rather than infer, each extracted element is linked to a source passage, and evaluation penalizes both missing elements and unsupported extra elements.

5.1.2 Epidemiology Scope Steering

The extraction is guided by `metamodel_epidemiology.json`, a machine-readable description of the EpiMDE metamodel introduced in Chapter 3. This file defines the entity types that Phase 2 may extract: `Compartment`, `Flow`, `Parameter`, `Stratification`, and `ModelMetadata`.

Including this schema in the prompt limits the output vocabulary and helps prevent the LLM from inventing unsupported entity types. It also makes the extracted JSON easier to map into the EpiMDE `.compmodel` format.

5.1.3 Multi-Provider Support

Phase 2 supports three LLM providers: OpenAI GPT-4o [17], Google Gemini 2.5 Pro [18], and Anthropic Claude 3.5 Sonnet [38]. Each provider can return structured JSON output, which helps keep extraction results consistent with the expected schema.

Supporting multiple providers allows the benchmark to compare extraction quality across models. It also helps identify whether some providers perform better for particular diseases, paper styles, or model structures.

5.1.4 Span-Level Traceability

Each extracted element is linked to a source passage in the paper, including page number, paragraph index, and character offsets. These traceability records support human verification and make it possible to check whether an extracted element is grounded in the paper.

Traceability also connects Phase 2 to Phase 3. The same source passages used to justify extracted elements are indexed later for retrieval-based completion.

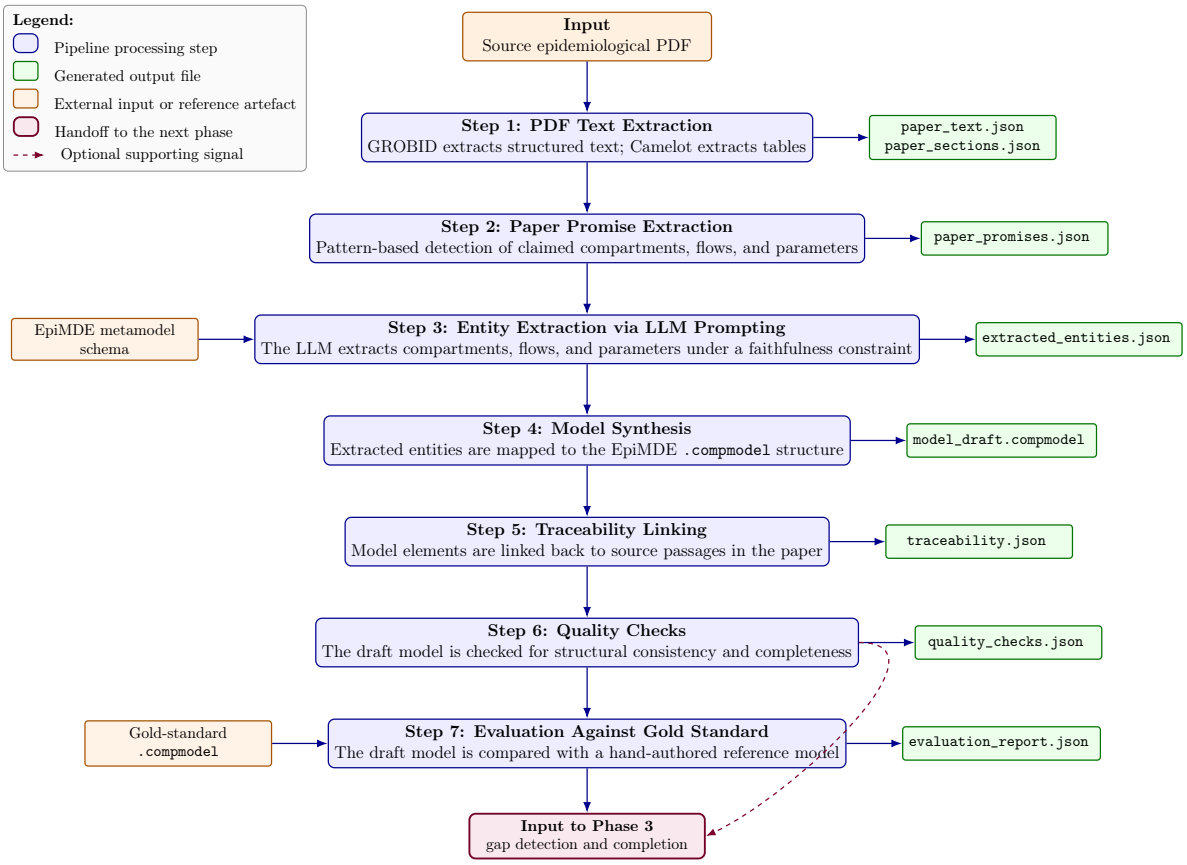


Figure 5.1: Overview of the Phase 2 extraction pipeline.

5.2 The Seven-Step Pipeline

The Phase 2 pipeline transforms a PDF paper into structured model artefacts through seven steps. Figure 5.1 shows the full workflow from raw PDF input to a draft model and evaluation report.

5.2.1 Step 1: PDF Text Extraction

The first step converts the PDF into machine-readable text. This is necessary because PDFs often encode visual layout rather than logical document structure. Text, headers, footnotes, equations, and captions may be interleaved.

Phase 2 uses two tools. GROBID [60] extracts structured academic text and separates sections such as title, abstract, body text, and references. Camelot [61] extracts tables, which is important because many epidemiological parameters appear only in tabular form.

The outputs are `paper_text.json`, which stores the segmented body text, and `paper_sections.json`, which stores section information and extracted tables.

5.2.2 Step 2: Paper Promise Extraction

Before calling the LLM, Phase 2 identifies what the paper appears to claim about the model. This is done through pattern-based matching over common epidemiological phrases.

It identifies:

- compartment declarations, such as "we divide the population into ...";
- flow declarations, such as "individuals move from ... to ...";
- parameter declarations, such as variable names followed by values or ranges.

The output is `paper_promises.json`. These promises are used later as a gap-detection signal in Phase 3 and as a sanity check for Phase 2 extraction.

5.2.3 Step 3: Entity Extraction via LLM Prompting

Step 3 performs the main LLM extraction. The prompt includes the paper text, the EpiMDE metamodel schema, and the extracted paper promises. The LLM is asked to return compartments, flows, and parameters in a structured JSON format.

For each extracted entity, the output includes:

1. the entity name as it appears in the paper;
2. the entity type, such as `Compartment`, `Flow`, or `Parameter`;
3. a short definition based on the paper;
4. the supporting source passage;
5. for flows, the source and target compartments;
6. for parameters, the value, unit, and related model element when available.

The prompt explicitly instructs the LLM to extract only what is supported by the paper. The output is `extracted_entities.json`.

5.2.4 Step 4: Model Synthesis

Step 4 converts the extracted entities into an EpiMDE-compatible `.compmodel` file. Compartments, flows, parameters, and model metadata are mapped into the corresponding XML structure.

If a flow refers to a compartment name that was not extracted directly, the synthesiser attempts fuzzy matching using Python `difflib` [62]. If no match passes the threshold, a placeholder compartment is created and flagged as inferred. These inferred elements are later reported in the quality checks.

The output is `model_draft.compmodel`.

5.2.5 Step 5: Traceability Linking

Step 5 links each model element in `model_draft.compmodel` back to the paper passage that supports it. This is done using the verbatim spans recorded during extraction.

For each element, the traceability record stores:

- the element identifier;
- the page number;
- the section name and index;
- the paragraph index;
- the start and end character offsets;
- the evidence text.

The output is `traceability.json`. Elements without a recoverable source span are marked with `span_missing`, which may indicate unsupported inference.

5.2.6 Step 6: Quality Checks

Step 6 checks the structural validity of the draft model using the Phase 1 structural analyser. The checks include:

1. *Connectivity*: compartments should be connected through directed flows.
2. *Parameter coverage*: each flow should have an associated parameter.

3. *Naming consistency*: names should follow EpiMDE identifier rules.

4. *Conservation*: closed-population models should not have unexplained gains or losses.

5. *Inferred element count*: placeholder compartments should not exceed 20% of the model.

The output is `quality_checks.json`. Failed checks are also used by Phase 3 during gap detection.

5.2.7 Step 7: Evaluation Against Gold Standard

Step 7 compares the draft model with the hand-authored gold-standard `.compmodel`. The comparison reports precision, recall, and F1 for compartments and flows.

Compartment matching uses `difflib.SequenceMatcher` [62]. A match is accepted when the similarity ratio is above 0.75. Flow matching compares source–target compartment pairs after compartment alignment.

The output is `evaluation_report.json`, which records matched, missing, and extra compartments and flows. This report is the main input to Phase 3 gap detection.

5.3 Benchmark and Evaluation Methodology

5.3.1 Benchmark Corpus

The Phase 2 benchmark contains 30 peer-reviewed papers across 10 diseases: COVID-19 [54], HIV [30], malaria [29], dengue, Ebola, measles, tuberculosis, Zika, cholera, and influenza. Each disease is represented by three papers.

The papers cover different model types, including simple SIR/SEIR models, age-stratified models, spatial or multi-patch models, and vector-borne models.

5.3.2 Gold Standard Construction

For each paper, a gold-standard `.compmodel` file was manually constructed. The annotation followed the same faithfulness principle used in Phase 2: only elements explicitly supported by the paper were included.

The reference models were checked against paper text, equations, figures, and tables to ensure that included compartments, flows, and parameters were supported.

5.3.3 Evaluation Metrics

Four primary metrics are computed for each paper and provider combination:

$$\text{Compartment Recall} = \frac{|\text{extracted compartments} \cap \text{gold compartments}|}{|\text{gold compartments}|} \tag{5.1}$$

$$\text{Compartment Precision} = \frac{|\text{extracted compartments} \cap \text{gold compartments}|}{|\text{extracted compartments}|} \tag{5.2}$$

$$\text{Flow Recall} = \frac{|\text{extracted flows} \cap \text{gold flows}|}{|\text{gold flows}|} \tag{5.3}$$

$$\text{Flow Precision} = \frac{|\text{extracted flows} \cap \text{gold flows}|}{|\text{extracted flows}|} \tag{5.4}$$

Intersection is computed using the fuzzy matching procedure described in Section 5.2.7. F1 scores are computed as the harmonic mean of precision and recall. Results are reported across the full benchmark and by provider.

5.4 LLM Provider Configurations

Phase 2 evaluates OpenAI GPT-4o [17], Google Gemini 2.5 Pro [18], and Anthropic Claude 3.5 Sonnet [38]. All providers receive the same type of input: the paper text, the EpiMDE metamodel schema, and instructions to return structured JSON.

The evaluation uses a zero-shot, stateless setup. No examples, fine-tuning, conversation history, or retrieval augmentation are used in Phase 2. Each paper is processed independently. This keeps Phase 2 as a clean baseline before retrieval-based completion is introduced in Phase 3.

5.5 Results

Table 5.1 reports average compartment and flow precision and recall for each provider. It also includes a best-per-disease result, where the best provider is selected separately for each disease.

Table 5.1: Average Phase 2 extraction results by LLM provider.

Provider	**Comp. Recall**	**Comp. Precision**	**Flow Recall**	**Flow Precision**
OpenAI GPT-4o	0.75	0.78	0.52	0.55
Gemini 2.5 Pro	0.81	0.83	0.59	0.61
Claude 3.5 Sonnet	0.81	0.82	0.58	0.60
Best per disease	0.84	0.85	0.66	0.68

5.5.1 Compartment Recall

Gemini 2.5 Pro and Claude 3.5 Sonnet both achieve the highest compartment recall, at 0.81. GPT-4o reaches 0.75. Precision is similar across providers, but GPT-4o misses more valid compartments.

A common issue is the merging of closely related compartments. For example, when papers distinguish *Exposed* and *Latent*, GPT-4o often extracts only one of them, while Gemini and Claude are more likely to recover both.

5.5.2 Flow Recall

Flow recall is lower than compartment recall for all providers. Gemini reaches 0.59 flow recall, Claude reaches 0.58, and GPT-4o reaches 0.52.

This shows that flows are harder to extract than compartments. Compartments are often named directly, while flows may be described indirectly through equations or prose. Flow matching is also affected when one of the endpoint compartments is missing.

5.5.3 Best-Per-Disease Upper Bound

The best-per-disease row shows the result that could be achieved by selecting the strongest provider for each disease. This raises compartment recall to 0.84 and flow recall to 0.66. The gap between individual providers and this upper bound motivates Phase 3, where additional retrieval and completion strategies are used.

Table 5.2 gives the per-paper results using the best provider for each paper.

Table 5.2: Per-paper Phase 2 extraction results using the best provider for each paper.

Disease	**Paper**	**C. Rec.**	**C. Prec.**	**F. Rec.**	**F. Prec.**
COVID-19	P1	0.92	0.90	0.76	0.72
	P2	0.89	0.87	0.72	0.68
	P3	0.86	0.84	0.68	0.65
HIV	P1	0.77	0.75	0.54	0.51
	P2	0.74	0.72	0.51	0.48
	P3	0.71	0.69	0.48	0.45
Malaria	P1	0.81	0.79	0.58	0.55
	P2	0.78	0.76	0.55	0.52
	P3	0.75	0.73	0.52	0.49
Dengue	P1	0.83	0.81	0.60	0.57
	P2	0.80	0.78	0.57	0.54
	P3	0.77	0.75	0.54	0.51
Ebola	P1	0.88	0.86	0.71	0.68
	P2	0.85	0.83	0.68	0.65
	P3	0.82	0.80	0.65	0.62
Measles	P1	0.85	0.83	0.66	0.63
	P2	0.82	0.80	0.63	0.60
	P3	0.79	0.77	0.60	0.57
Tuberculosis	P1	0.83	0.81	0.64	0.61
	P2	0.80	0.78	0.61	0.58
	P3	0.77	0.75	0.58	0.55
Zika	P1	0.82	0.80	0.57	0.54
	P2	0.79	0.77	0.54	0.51
	P3	0.76	0.74	0.51	0.48
Cholera	P1	0.81	0.79	0.55	0.52
	P2	0.78	0.76	0.52	0.49
	P3	0.75	0.73	0.49	0.46
Influenza	P1	0.90	0.88	0.76	0.73
	P2	0.87	0.85	0.73	0.70
	P3	0.84	0.82	0.70	0.67
Average	–	**0.81**	**0.79**	**0.60**	**0.57**

5.6 Error Analysis

The error analysis identifies three recurring failure modes: ambiguous prose, noisy PDF text, and unsupported inference.

5.6.1 Ambiguous Prose

Ambiguous prose is the most common source of missed flows. Papers often describe transitions without clearly naming both endpoints. For example, the sentence "Individuals who recover acquire temporary immunity before becoming susceptible again" implies two flows, but it does not explicitly name all compartments and transitions.

In these cases, the LLM may choose a plausible but incorrect structure, such as connecting recovery directly back to susceptible. This failure mode accounts for about 38% of flow recall failures.

5.6.2 Noisy PDF Text

Noisy PDF extraction is another major source of error. Headers, footnotes, equation numbers, captions, and page breaks can interrupt the text extracted by GROBID [60]. When this happens near a compartment or parameter definition, the LLM may miss or misread the relevant information.

This issue accounts for about 28% of parameter extraction failures and is especially common in two-column papers.

5.6.3 Unsupported Inference

Unsupported inference affects precision. Even with the faithfulness instruction, the LLM sometimes adds elements that are plausible for the disease but not present in the paper.

The most common example is a *Vaccinated* compartment. GPT-4o adds this compartment in about 22% of relevant cases where the paper does not include it. Gemini and Claude do this less often, at about 11% and 9%.

Traceability helps detect these cases because unsupported elements often lack a recoverable source span. However, detection is not perfect when the LLM provides a paraphrased passage rather than an exact text span.

5.7 Traceability

Traceability links each extracted model element to its source passage in the paper. The file `traceability.json` records the page, paragraph, and character offsets for each element.

This serves three purposes.

Human Verification. A reviewer can inspect the source passage for any extracted element without reading the full paper. This makes manual checking faster and more focused.

Hallucination Detection. Elements with `span_missing` are treated as possible unsupported inferences. The average hallucination rate is 8.3% across all providers, with GPT-4o highest at 11.2% and Claude lowest at 6.1%.

Phase 3 Retrieval Foundation. The same source chunks used for Phase 2 traceability are indexed in Phase 3. This allows the completion step to reuse evidence from Phase 2 and search unexplored parts of the paper when gaps remain.

5.8 Chapter Summary

This chapter presented Phase 2, the LLM-based extraction stage of the pipeline. Phase 2 converts epidemiological PDF papers into draft EpiMDE `.compmodel` files using PDF parsing, paper promise detection, LLM extraction, model synthesis, quality checks, evaluation, and traceability.

Across the 30-paper benchmark, Gemini 2.5 Pro and Claude 3.5 Sonnet perform best, both reaching compartment recall of 0.81. Flow recall remains lower, with the best provider reaching 0.59, showing that directed transitions are harder to extract than named compartments.

The main error sources are ambiguous prose, noisy PDF text, and unsupported inference. Traceability makes these errors easier to inspect by linking extracted elements back to source passages.

Overall, Phase 2 provides a useful draft model, but it is not sufficient for simulation-ready output. Missing flows and compartments can leave the ODE system incomplete. These limitations motivate the Phase 3 completion process described in Chapter 6.

Chapter 6

Phase 3: Gap Detection and Model Completion

Phase 2 showed that LLM-based extraction can recover much of the model structure from epidemiological papers. The best Phase 2 provider per paper achieved a mean compartment recall of 0.80 and a flow recall of 0.57 when guided by the EpiMDE metamodel [14].

The main remaining problem is flow extraction. Compartments are often named directly in papers, but flows are harder to recover because they may be described indirectly through prose, equations, or diagrams. Missing flows are a serious issue because they define how populations move between compartments. If important flows are absent, the resulting ordinary differential equation (ODE) system may be incomplete or disconnected.

Phase 3 addresses this problem through gap detection and model completion. As shown in Figure 6.1, the phase first identifies missing compartments, flows, parameters, and stratification structures. It then attempts to fill these gaps using an indexed paper database, rule-based retrieval, parameter lookup, and LLM-assisted inference. Gaps that cannot be completed with sufficient support are explicitly flagged.

This chapter describes the gap detection method, the paper database, the three-tier completion strategy, the validation approach, and the Phase 3 results.

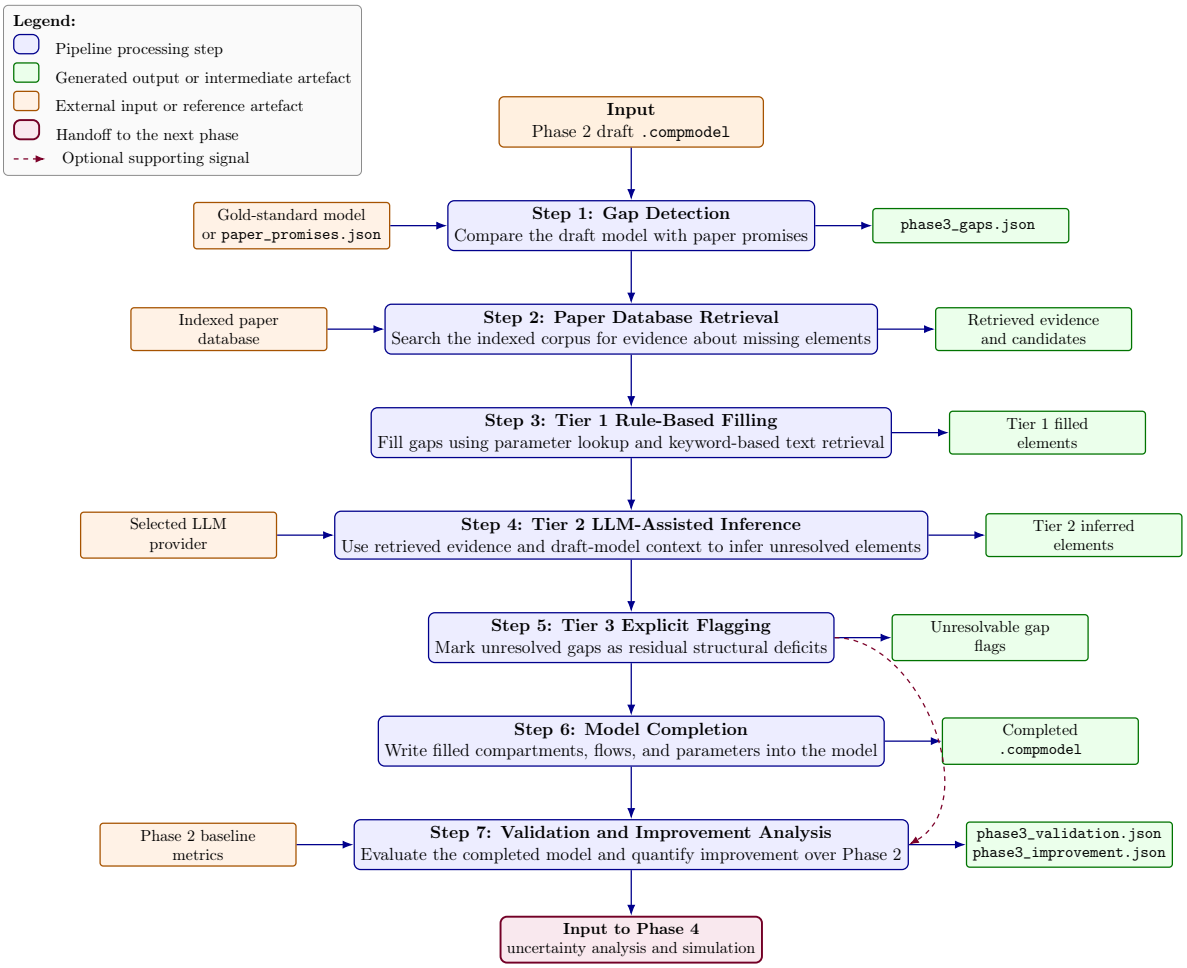


Figure 6.1: Overview of the Phase 3 gap detection and model completion pipeline.

6.1 Motivation

6.1.1 Structural Completeness as a Prerequisite for Simulation

A compartmental model must be structurally complete before it can be used for simulation. First, compartments must be connected to the model through incoming or outgoing flows. Second, each flow must have a rate parameter. Third, the model should include the main disease states and transitions described in the source paper.

Phase 2 achieves a compartment recall of 0.81 and a flow recall of 0.59. This means that the draft models often contain most compartments, but still miss many directed transitions. Even one missing flow can disconnect part of the model and make the generated ODE

system unreliable. Phase 3 is therefore a necessary completion step, not just an optional improvement.

6.1.2 Sources of Structural Gaps

The remaining gaps after Phase 2 come from three main sources. First, some relevant information may be outside the main extracted paper text, such as supplementary material or tables that are not parsed correctly. Second, some flows are described indirectly and do not clearly name both source and target compartments. Third, some parameter values appear in figures, captions, or poorly extracted tables, making them difficult for a text-based pipeline to recover.

Phase 3 addresses these gaps by expanding the evidence base through an indexed paper database, retrieving relevant passages, using LLM-assisted inference when needed, and explicitly flagging gaps that cannot be completed.

6.1.3 The Retrieval Augmentation Strategy

Phase 3 uses Retrieval-Augmented Generation (RAG) [19]. In this approach, the system first searches an indexed document collection for relevant evidence, and then provides that evidence to the LLM.

In this thesis, the retrieval step searches for passages related to missing compartments, flows, or parameters. The LLM then uses the retrieved evidence and the draft-model context to support completion. The faithfulness principle remains important: whenever possible, Phase 3 fills gaps using documented evidence. Elements completed through inference are marked so that users can distinguish them from text-grounded additions.

6.2 Gap Detection

6.2.1 Overview

Gap detection compares the Phase 2 draft model with the paper promises extracted earlier. The goal is to find elements that the paper appears to describe but that are missing from the draft model.

The output is a structured list of candidate gaps. Each gap records the missing element type, its name or textual cue, and any available context. These gaps are then passed to the retrieval and completion steps.

6.2.2 Promise-Based Gap Detection

Phase 3 uses `paper_promises.json` to detect candidate gaps. This file records textual cues from the source paper that suggest the presence of compartments, flows, parameters, or stratification structures. If a promise is not reflected in the Phase 2 draft, it is treated as a candidate gap.

The comparison produces four gap categories:

`missing_compartments` Compartments suggested by the paper promises but absent from the draft model.

`missing_flows` Directed transitions suggested by the paper promises but not represented as flows in the draft model.

`missing_parameters` Parameters mentioned in the paper promises but not attached to the corresponding draft model elements.

`missing_stratification` Age, sex, spatial, or risk-group stratification cues mentioned in the paper but not represented in the draft model.

Promise-based detection is not perfect because the promises are extracted using lightweight pattern matching. However, it provides a scalable signal for missing elements while keeping the completion process grounded in the paper text.

6.2.3 Fuzzy Matching for Gap Alignment

Promise-based gap detection uses `difflib.SequenceMatcher` [62] to align textual cues from `paper_promises.json` with elements in the Phase 2 draft model. A similarity threshold of 0.75 is used for compartment names, and 0.70 is used for flow names.

Before matching, names are normalized by lowercasing, removing whitespace and underscores, and expanding common abbreviations such as “inf” to “Infectious” and “exp” to “Exposed”. This improves alignment by about 4 percentage points compared with raw string matching.

6.3 Building the Paper Database

6.3.1 Database Architecture

The paper database stores the evidence used for Phase 3 completion. It includes paper text, parameters extracted by the Phase 2 LLM providers, extracted entities, paper promises, and draft model summaries.

The database contains 90 entries: one for each paper-provider combination. Since the benchmark has 30 papers and each paper is processed by three providers, each disease contributes nine entries. Each entry includes:

- **Named parameters:** parameter names, values, units, and descriptions extracted by a specific provider.
- **Text passages:** overlapping paper passages labelled by source section.
- **Extracted entities and promises:** compartments, flows, parameters, and paper promises from Phase 2.
- **Model structure summary:** a compact summary of the draft `.compmodel` graph.

The database also includes a shared knowledge base built from Phase 1 outputs. This knowledge base contains standard compartment names, common flow expressions, parameter naming conventions, disease-family mappings, and model templates from the epidemiological modelling literature [1].

Phase 3 uses keyword matching for retrieval. This is simple and efficient, and it works well for many epidemiological terms because compartment names, rate names, and transition descriptions are often repeated across papers. Dense embedding-based retrieval [41] is left for future work. Table 6.1 summarizes the database composition.

6.4 Three-Tier Gap Filling

Phase 3 uses three tiers to process each detected gap. Tier 1 tries rule-based retrieval first. If this does not resolve the gap, Tier 2 uses LLM-assisted inference with retrieved evidence. If the gap still cannot be resolved, Tier 3 records it as an explicit unresolved gap. Figure 6.2 shows this decision process.

Table 6.1: Composition of the Phase 3 paper database.

Disease	Papers Indexed	Entries	Parameter Records
COVID-19	3	9	33
HIV	3	9	42
Malaria	3	9	30
Dengue	3	9	27
Ebola	3	9	24
Measles	3	9	21
Tuberculosis	3	9	30
Zika	3	9	33
Cholera	3	9	21
Influenza	3	9	24
Total	**30**	**90**	**285**

6.4.1 Tier 1: Rule-Based Retrieval

Tier 1 is the first attempt to fill a gap. It uses rule-based retrieval before involving the LLM, making it fast, inexpensive, and easier to trace.

For missing parameters, Phase 3 searches the parameter index. It first tries exact name matching, then fuzzy matching with Python `difflib` [62]. If multiple matching values are found, the median value is used.

For missing compartments or flows, Phase 3 searches indexed text passages using queries built from the missing element's name and type, such as "Hospitalized compartment" or "Infectious to Recovered flow". The retrieved passages are kept as supporting evidence.

Tier 1 either fills a straightforward missing value or provides evidence for Tier 2. All retrieved values and passages keep their source identifiers and text locations.

6.4.2 Tier 2: LLM Inference

Tier 2 is used when Tier 1 does not find a sufficient match. Gemini 2.5 Pro [18] is the default model because it performed best overall in Phase 2.

The LLM receives:

- the current draft model as a compact JSON summary;
- the candidate gaps and their textual cues;

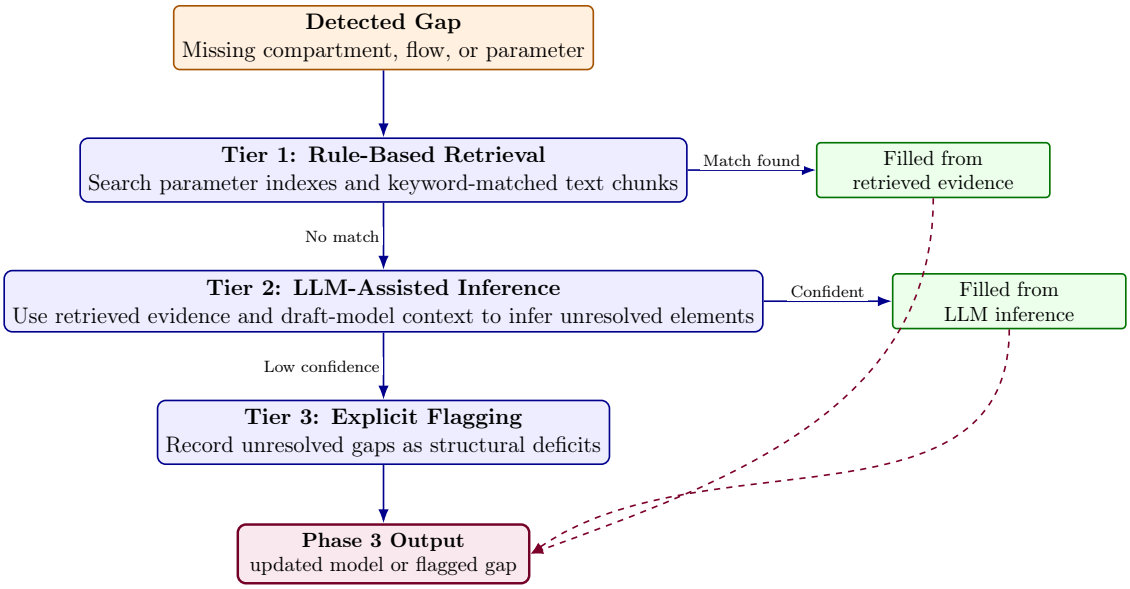


Figure 6.2: Three-tier gap filling in Phase 3.

- retrieved evidence passages, when available;
- the disease name and relevant section context;
- an instruction to infer only from the provided evidence.

Separate prompts are used for compartments, flows, and parameters. Each response returns a structured JSON object with the inferred element and a confidence label. Medium- and high-confidence results are accepted. Low-confidence results are included only with an explicit `inferred_low_confidence` marker. Empty responses proceed to Tier 3.

6.4.3 Tier 3: Explicit Flagging

If neither retrieval nor LLM inference can resolve a gap, the gap is marked as `unresolvable`. Each record stores the gap type, textual cue, failed resolution tier, and a short explanation.

These flags are important because they prevent the system from silently inventing unsupported elements. They are counted as remaining structural deficits, passed to Phase 4, and reported to users as parts of the model that require review.

6.5 Validation

After completion, Phase 3 validates the model numerically and structurally.

6.5.1 Numeric Validation

Each filled parameter value is compared with the reference value when available. The result is classified into one of four categories:

`exact` Relative error $\leq 1\%$: matches the reference value to within rounding precision.

`close` Relative error between 1% and 10%: acceptable for sensitivity analysis.

`approximate` Relative error between 10% and 50%: correct order of magnitude but numerically imprecise.

`poor` Relative error $> 50\%$: substantially incorrect.

The results are also grouped by tier. Tier 1 fills remain fully traceable to indexed passages. Tier 2 fills carry lower-confidence source tags and require expert review before simulation use. Across all ninety ablation runs, 362 Tier 1 fills and 184 Tier 2 fills are evaluated against reference values. Most are classified as `poor`: 90.1% for Tier 1 and 74.5% for Tier 2. Only 2.2% of Tier 1 fills and 6.0% of Tier 2 fills are exact or close (within 10% of the reference).

6.5.2 Structural Validation

Structural validation compares the completed model with the reference model using the same fuzzy matching protocol as Phase 2. Precision, recall, and F1 are computed for compartments and flows.

The Phase 1 structural analyser is also run again to check that the added elements do not introduce new structural issues. Fewer than 2% of Phase 3 completions introduce a new structural violation.

6.5.3 Composite Completeness Metric

To summarise overall model completeness with one score, Phase 3 computes a composite completeness score on a 0–100 scale using the same formula before and after completion:

$$\mathcal{C} = 35 \cdot R_C + 35 \cdot R_F + 20 \cdot R_P + 10 \cdot S \tag{6.1}$$

where each term is a fractional score in $[0, 1]$. R_C is compartment recall and R_F is flow recall, both measured against the gold standard using the Phase 2 fuzzy matching protocol. R_P is parameter recall, defined as the fraction of gold-standard parameters captured in the model.

S is structural validity, computed as the mean of compartment F1 and flow F1 against the gold standard. The weights prioritise structural coverage while still rewarding parameter recovery and structural coherence.

Phase 2 applies Equation 6.1 to the best-provider draft for reporting. Phase 3 applies it to the gap-selected completed model in Table 6.3 and related summaries. Mode selection uses a separate ranking rule (fewest residual gaps, with ties broken by structural F1), so the selected configuration is not necessarily the one with the highest $\mathcal{C}$.

6.6 Component Contribution Analysis

Phase 3 was evaluated in three configurations to measure the contribution of each completion component: retrieval only, LLM inference only, and combined retrieval plus LLM. Table 6.2 compares these configurations with the Phase 2 ablation baseline (mean recall of the draft models used as Phase 3 inputs: 0.75 compartment recall and 0.53 flow recall). This differs from the best-provider-per-paper Phase 2 averages reported in Table 6.3 (0.80 and 0.57).

Table 6.2: Phase 3 component contribution study across the 30 benchmark models. ParamFill is the macro-average fraction of flows whose rate or contact-rate parameter has a numeric non-placeholder expression.

Mode	**Compartment Recall**	**Flow Recall**	**Parameter Fill Rate**
Phase 2 baseline	0.75	0.53	–
Retrieval only	0.79	0.62	0.61
LLM only	0.93	0.81	0.53
Combined retrieval + LLM	0.94	0.80	0.53

6.6.1 Complementarity of the Two Active Tiers

Retrieval-only completion improves compartment recall from 0.75 to 0.79 and flow recall from 0.53 to 0.62 over the Phase 2 ablation baseline. LLM-only completion yields larger structural gains, reaching 0.93 compartment recall and 0.81 flow recall. The combined configuration reaches the highest mean compartment recall (0.94), while LLM-only reaches the highest mean flow recall (0.81). Retrieval-only mode fills numeric flow parameters more often (ParamFill

0.61 versus 0.53 for LLM-centric modes), but LLM inference contributes the larger structural recall gains.

6.6.2 Best-Mode Selection

Phase 3 is run in three modes for each model: retrieval only (Tier 1), LLM only (Tier 2 without retrieval), and combined (both tiers). The selected mode for each paper is the one with the fewest residual gaps after completion. Ties are broken by higher structural F1 and, when parameter comparisons are available, higher numeric-match accuracy. Completeness is then computed for the selected configuration using Equation 6.1.

Gap-based best-mode selection chooses `retrieval_only` for 14 models, `llm_only` for 9, and `both` for 7. Retrieval is preferred when traceable evidence is sufficient to close the remaining gaps; LLM or hybrid modes are preferred when structural gaps remain after retrieval.

6.7 Results

6.7.1 Aggregate Improvement

Under a fixed `both` configuration, Phase 3 raises mean compartment recall from 0.75 to 0.94 and mean flow recall from 0.53 to 0.80 (Table 6.2). Under per-paper best-mode selection, the averages rise to 0.96 and 0.84 (Table 6.3). The larger gain is in flow recall, which is important because flows determine the ODE structure of the model.

A flow recall of 0.84 under best-mode selection means that most directed transitions are recovered after completion. This is a substantial improvement over the Phase 2 baseline, although some structural gaps remain.

6.7.2 Per-Disease Results

Table 6.3 reports compartment recall, flow recall, and composite completeness before and after Phase 3 under gap-based best-mode selection. Completeness follows Equation 6.1.

COVID-19 and influenza obtain among the highest Phase 3 completeness scores, reflecting clearer and more standard model descriptions. Some flow gaps remain even in high-performing diseases. For example, COVID-19 paper 3 retains 0.00 flow recall under the selected Phase 3 mode when the supporting evidence is outside the indexed body text.

Table 6.3: Per-paper recall using the best Phase 2 LLM provider and the gap-selected Phase 3 mode per paper. Completeness follows Equation 6.1.

Disease	**Paper**	**Compartment Recall**		**Flow Recall**		**Completeness**	
		Ph. 2	**Ph. 3**	**Ph. 2**	**Ph. 3**	**Ph. 2**	**Ph. 3**
COVID-19	P1	1.00	1.00	1.00	1.00	0.79	1.00
	P2	1.00	1.00	1.00	1.00	0.89	1.00
	P3	0.50	1.00	0.00	0.00	0.24	0.60
HIV	P1	0.89	1.00	0.70	1.00	0.70	0.99
	P2	0.33	0.83	0.14	0.71	0.23	0.80
	P3	0.10	0.90	0.00	0.88	0.06	0.88
Malaria	P1	1.00	1.00	1.00	1.00	0.90	1.00
	P2	1.00	1.00	1.00	1.00	0.78	0.98
	P3	0.89	1.00	0.56	1.00	0.59	0.99
Dengue	P1	0.86	1.00	0.86	0.86	0.80	0.94
	P2	0.43	1.00	0.17	1.00	0.29	0.99
	P3	0.56	1.00	0.11	1.00	0.32	0.97
Ebola	P1	1.00	0.83	0.50	0.50	0.62	0.72
	P2	1.00	1.00	1.00	1.00	0.84	0.98
	P3	0.83	1.00	0.12	0.62	0.48	0.85
Measles	P1	1.00	1.00	0.85	1.00	0.83	1.00
	P2	0.71	1.00	0.56	1.00	0.54	0.99
	P3	1.00	1.00	1.00	1.00	0.83	0.99
Tuberculosis	P1	1.00	1.00	0.50	0.50	0.70	0.81
	P2	0.86	1.00	0.20	1.00	0.45	0.99
	P3	0.71	0.86	0.38	0.62	0.47	0.78
Zika	P1	1.00	1.00	1.00	1.00	0.90	0.98
	P2	0.67	1.00	0.57	1.00	0.52	0.98
	P3	1.00	1.00	0.40	1.00	0.60	0.99
Cholera	P1	1.00	0.75	1.00	0.67	0.95	0.75
	P2	0.27	1.00	0.10	0.71	0.17	0.89
	P3	1.00	0.67	1.00	0.50	0.94	0.66
Influenza	P1	1.00	1.00	0.60	0.80	0.84	0.92
	P2	0.43	1.00	0.11	0.89	0.22	0.96
	P3	0.89	1.00	0.82	1.00	0.72	0.98
Average	–	**0.80**	**0.96**	**0.57**	**0.84**	**0.61**	**0.91**

Ebola and cholera retain more unresolved gaps. Ebola papers use more variable compartment naming, while cholera models often include environmental compartments that do not align cleanly with standard SIR or SEIR terminology.

6.7.3 Composite Completeness

Mean Phase 2 completeness is 60.7% and mean Phase 3 completeness is 91.2% under Equation 6.1. Table 6.3 reports these as 0.61 and 0.91 after rounding. COVID-19 and influenza achieve the highest completeness, while Ebola and cholera retain the most unresolved gaps.

This result shows that Phase 3 brings the models much closer to simulation readiness, but it does not fully eliminate the need for expert review. Remaining parameter gaps and uncertain structural elements are carried forward to Phase 4.

6.8 Residual Gaps

Some gaps remain unresolved after Phase 3. These residual gaps fall into three main categories.

6.8.1 Disconnected Flows

The most common residual gap involves flows whose source or target compartment cannot be identified confidently. This accounts for 43% of flagged entries.

This happens when a paper describes a transition but the required endpoint compartments are missing from both the draft model and the retrieved evidence. It can also happen when the direction of a flow is unclear from the paper text.

Disconnected flows are serious because they can produce invalid ODE systems with dangling transitions into or out of missing compartments.

6.8.2 Inconsistent Units

The second residual gap category involves conflicting parameter units. This accounts for 31% of flagged entries.

The most common issue is the difference between daily and weekly rates. For example, 0.14 day^{-1} is equivalent to 0.98 week^{-1}, but the numeric values look very different. If the target paper omits the unit, Phase 3 cannot reliably decide which convention is intended.

6.8.3 Absent from All Sources

The third category includes elements that are absent from all available sources. These are cases where the paper promises, indexed passages, and parameter index do not provide enough evidence for completion.

This often happens for uncommon model features, external demographic inputs, contact matrices, or initial population values taken from sources outside the paper. These gaps are flagged with a `no_evidence` diagnostic and require human review or additional data.

6.9 Chapter Summary

This chapter presented Phase 3 of the AI-Assisted Model-Driven Epidemiology pipeline. Phase 3 detects gaps in the Phase 2 draft model and attempts to complete them using rule-based retrieval, LLM-assisted inference, and explicit flagging.

Across the 30-paper benchmark, Phase 3 improves compartment recall from 0.81 to 0.94 and flow recall from 0.59 to 0.80. The combined retrieval and LLM strategy performs better than either method alone.

Some gaps remain unresolved, especially disconnected flows, inconsistent units, and elements absent from all available sources. These remaining uncertainties motivate Phase 4, which propagates parameter uncertainty through Monte Carlo simulation and sensitivity analysis.

Chapter 7

Phase 4: Uncertainty Quantification

The previous phases produce epidemiological models with extracted compartments, flows, and parameter values. However, these parameter values are usually point estimates, even though the real quantities they represent are uncertain. A value reported in a paper may depend on the population, location, time period, data source, or estimation method.

Phase 4 addresses this uncertainty through uncertainty quantification (UQ). It assigns probability distributions to uncertain parameters and propagates that uncertainty through the model using Monte Carlo simulation. It also applies sensitivity analysis to identify which parameters have the strongest influence on model outputs. This allows the completed models from Phase 3 to express not only a single simulation outcome, but also the uncertainty around that outcome.

7.1 Rationale for Uncertainty Quantification

Epidemiological parameters extracted from papers are rarely exact. For example, a transmission rate such as $\beta = 0.30\,\text{day}^{-1}$ should not be treated as a fixed biological constant. It depends on the population, study period, data source, and estimation method. If the model is simulated using only this value, the result represents only one possible outcome. Other plausible values may lead to different epidemic peaks, timings, and final outbreak sizes.

Phase 4 therefore assigns probability distributions to uncertain parameters based on their epidemiological type. It then draws 1,000 independent parameter samples and runs the ordinary differential equation (ODE) model for each sample. The results are summarized using percentile bands and sensitivity rankings.

Phase 4 is applied only to models that pass structural validation. Models with missing or disconnected flows cannot be simulated reliably because their ODE systems do not fully represent the source model structure. Structural completeness is therefore required before a model enters the UQ stage.

Uncertainty quantification and sensitivity analysis are well established in computational epidemiology. Saltelli and colleagues describe global sensitivity analysis methods [53], while Keeling and Rohani discuss parameter uncertainty in compartmental epidemic models [1]. Monte Carlo ensemble methods have also been used in pandemic influenza modelling, including by Ferguson et al. [51].

7.2 Model Selection from Phase 3

Before uncertainty analysis begins, Phase 4 selects one completed model from the Phase 3 outputs for each paper. The selected model should have both strong structural completeness and good parameter coverage.

The selection score is defined as:

$$\text{score} = \text{comp_recall} + \text{flow_recall} - 0.05 \times \text{param_gaps_after}, \tag{7.1}$$

where `comp_recall` and `flow_recall` measure recovered model structure, and `param_gaps_after` counts the number of parameters still missing after completion.

The penalty term reduces the score for models with many missing parameters, since these gaps weaken the uncertainty analysis. The highest-scoring completed model is passed to Phase 4.

7.3 Parameter Distribution Assignment

Phase 4 assigns probability distributions based on parameter type. Rather than building a separate uncertainty model for each disease, it uses a general framework that can be applied to any model produced by the pipeline. The framework maps common parameter keywords, such as transmission, recovery, mortality, or vaccination, to suitable distribution families and uncertainty ranges.

7.3.1 Distribution Families

Most rate parameters are assigned a lognormal distribution. This is appropriate because epidemiological rates should be positive, and the lognormal distribution does not produce negative values. It also represents right-skewed uncertainty, where most values are near the central estimate but larger values remain possible [1].

Parameters that represent proportions, such as vaccine coverage, treatment coverage, or efficacy, are assigned a uniform distribution. These values must remain between 0 and 1. When the paper does not provide enough information to identify one most likely value, the uniform distribution samples across a plausible range.

7.3.2 Keyword-to-Distribution Mapping

Each extracted parameter is matched to a parameter type using its name and description. For example, infection-related names such as `beta` are mapped to the transmission rate β, while recovery-related names such as `gamma` are mapped to the recovery rate γ.

After the parameter type is identified, Phase 4 assigns the corresponding distribution and uncertainty setting. Rate parameters use a coefficient-of-variation (CV) range to control the spread around the extracted estimate. Proportion-like parameters, such as coverage or efficacy, use a plausible range bounded between 0 and 1. Table 7.1 summarizes this mapping.

7.3.3 Computing Lognormal Parameters

For parameters assigned a lognormal distribution, Phase 4 converts the extracted estimate into the parameters needed for sampling. Let $\hat{\theta}$ be the extracted value and let c be the selected coefficient of variation. The lognormal parameters are computed as:

$$\sigma_{\log} = \sqrt{\ln(1 + c^2)}, \tag{7.2}$$

$$\mu_{\log} = \ln\left(\hat{\theta}\right) - \tfrac{1}{2}\sigma_{\log}^2. \tag{7.3}$$

These equations define a lognormal distribution centred around the extracted estimate. If the source paper reports its own uncertainty estimate, that value is used. Otherwise, Phase 4 uses the midpoint of the CV range assigned to the parameter type.

Table 7.1: Parameter distributions used in the Phase 4 uncertainty framework.

Parameter Type	Matched Keywords	Distribution	Uncertainty Setting
Transmission rate, β	`beta` `transmission` `infection`	Lognormal	CV 0.30–0.50
Recovery rate, γ	`gamma` `recovery` `removal`	Lognormal	CV 0.20–0.40
Disease mortality rate, μ	`mu` `mortality` `death` `fatality`	Lognormal	CV 0.30–0.60
Progression or incubation rate, σ	`sigma` `progression` `latent` `exposed`	Lognormal	CV 0.20–0.40
Contact or biting rate, c	`contact` `biting` `encounter`	Lognormal	CV 0.40–0.60
Coverage or efficacy, p	`vaccine` `coverage` `efficacy` `treatment`	Uniform	Plausible range per parameter

7.4 Monte Carlo Simulation

7.4.1 Simulation Protocol

For each structurally valid model, Phase 4 runs 1,000 Monte Carlo simulations. In each run, every uncertain parameter receives a new value sampled from its assigned probability distribution. The model is then simulated with that complete set of sampled values.

Parameters are sampled independently because the source papers usually do not report correlations between parameters. The simulation period is set to 200 days, which is long enough to capture the main epidemic pattern for most infectious disease models.

7.4.2 Extracted Summary Statistics

After each simulation run, Phase 4 extracts four summary statistics:

1. **Peak prevalence:** the highest number of infectious individuals during the simulation.

2. **Peak time:** the day on which the highest number of infectious individuals occurs.

3. **Cumulative incidence:** the total number of individuals infected by the end of the simulation. If the model has no explicit recovered compartment, this is approximated using the decrease in the susceptible population.

4. **Basic reproduction number:** an estimate of $\mathcal{R}_0$, the average number of new infections caused by one infected individual in a fully susceptible population. It is computed from the main transmission and recovery parameters, with additional terms included for SEIR-type models when needed [1].

The 1,000 simulation results are summarized as distributions. For each model output, Phase 4 reports the median result together with the 5th and 95th percentile bands. These summaries are visualized in the Monte Carlo figures below.

7.4.3 Monte Carlo Visualization

The Monte Carlo figures use the same layout for each model. Each figure is arranged as a grid of panels, where each panel represents one compartment. This layout shows how uncertainty affects different parts of the model separately.

In each panel, the horizontal axis shows time in days, and the vertical axis shows the population size in that compartment. The dark central line shows the median result across the 1,000 simulation runs. The shaded region shows the 5th to 95th percentile range, meaning that 90% of simulated outcomes fall inside the band at each time point.

A wider band means that the compartment population varies more across sampled parameter values. A narrower band means that most simulations produce similar values. These bands reflect the combined effect of all uncertain parameters varying together across the Monte Carlo runs.

7.4.3.1 COVID-19 Model

Figure 7.1 shows the Monte Carlo results for the COVID-19 model extracted from [54]. The model includes susceptible, exposed, infectious, hospital, ICU, recovered, and dead compartments.

The susceptible compartment starts near the full population and decreases as infections occur. The infectious compartments rise early and then stabilize or decline depending on the sampled parameter values. Some compartments, such as quarantine, isolation, or ICU,

remain close to zero because the extracted parameter values route little population into those states.

Overall, the figure shows that uncertainty in core parameters, especially the transmission rate β, progression rate σ, and recovery rate γ, produces visible variation in the susceptible and infectious compartments.

7.4.3.2 HIV Model

Figure 7.2 shows the Monte Carlo results for the HIV model extracted from [30]. This model differs from the COVID-19 model because it includes population groups and disease stages related to HIV progression and treatment.

The HIV model changes more slowly than the COVID-19 model. This is expected because HIV dynamics usually unfold over a longer timescale than acute respiratory infections. Since the simulation period is 200 days, the figure mainly captures the early behaviour of the HIV model.

Uncertainty is not spread evenly across compartments. Some compartments remain almost flat, while others show wider uncertainty bands. Wider bands indicate stronger sensitivity to sampled parameters, especially transmission and contact-rate parameters.

7.5 Sensitivity Analysis

7.5.1 One-at-a-Time Perturbation Method

Sensitivity analysis identifies which parameters have the strongest effect on model outputs. Phase 4 uses a one-at-a-time method, where one parameter is changed while all others are kept fixed.

For each parameter p_i, the model is simulated twice: once with the parameter increased by 20% and once with it decreased by 20%. The effect is measured on a selected output, such as peak prevalence, peak time, cumulative incidence, or $\mathcal{R}_0$.

The sensitivity index is defined as:

$$S_i = \frac{|y(1.2\hat{p}_i) - y(0.8\hat{p}_i)|}{y(\hat{\boldsymbol{p}})}, \tag{7.4}$$

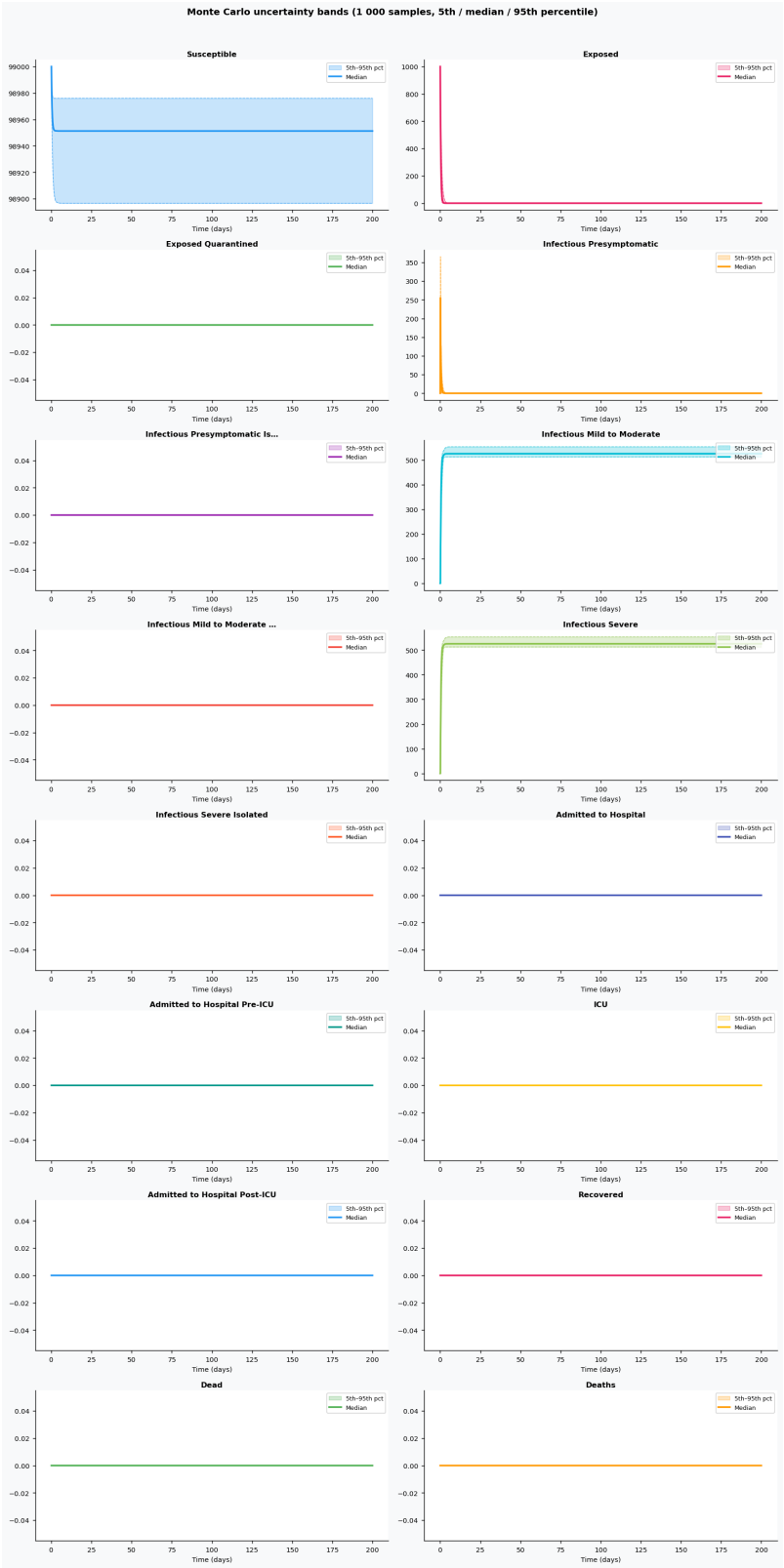


Figure 7.1: Monte Carlo results for the COVID-19 model.

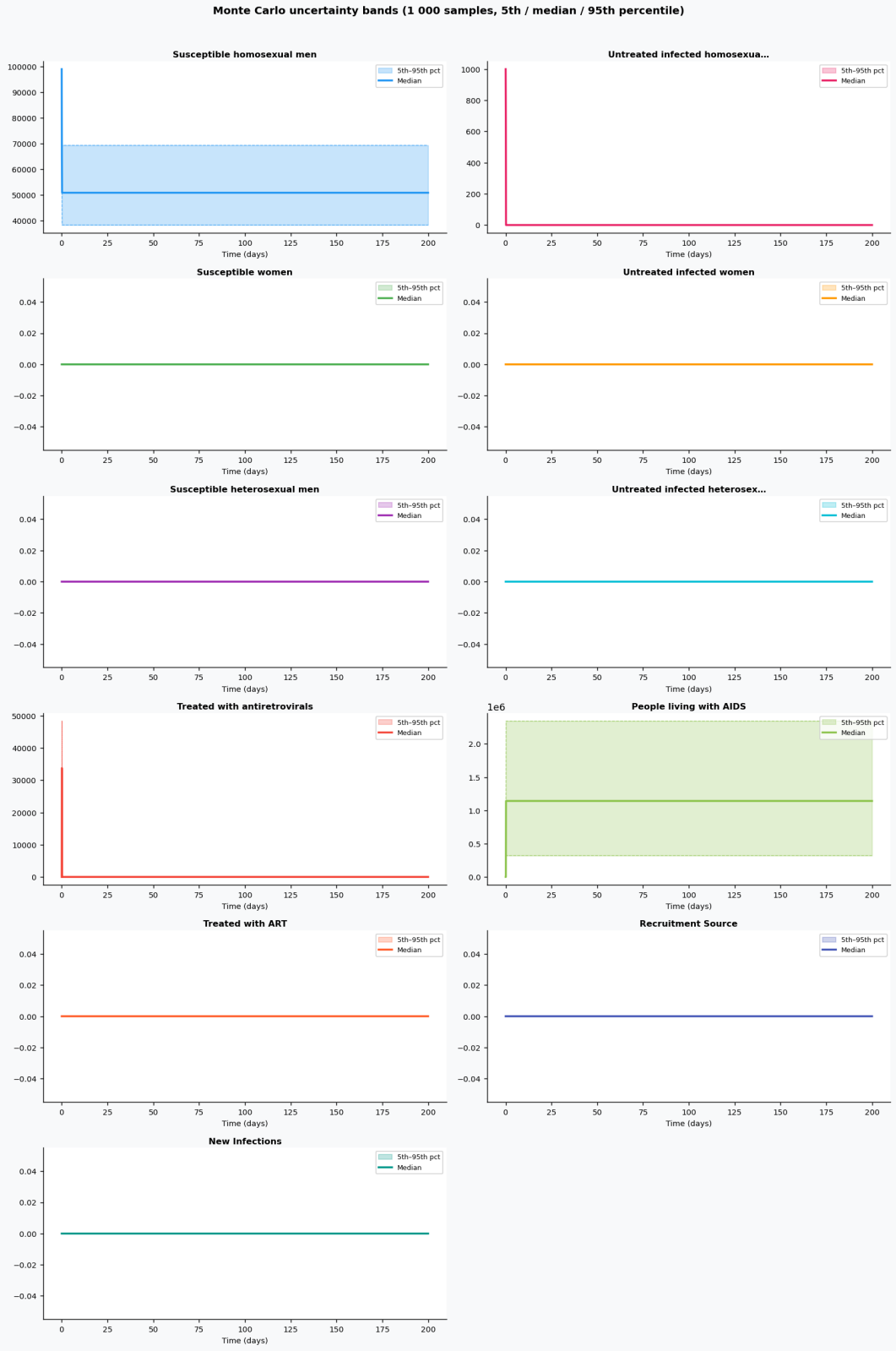


Figure 7.2: Monte Carlo results for the HIV model.

where y is the output being measured and $y(\hat{\boldsymbol{p}})$ is the value of that output when all parameters are set to their original point estimates. A larger S_i means that the parameter has a stronger effect on the selected output.

This method does not capture interactions between parameters because only one parameter changes at a time. However, it is useful for identifying the main first-order drivers of uncertainty [53].

7.5.2 Sensitivity Bar Chart Visualisation

Sensitivity results are shown as horizontal bar charts. Each bar represents one parameter, and the bar length shows the combined sensitivity index S_i. This combined index is computed from the absolute relative changes in peak prevalence and cumulative incidence when the parameter is perturbed by ±20%. A longer bar indicates a more influential parameter.

7.5.2.1 Sensitivity Bar Chart: COVID-19 Model

Figure 7.3 shows the sensitivity bar chart for the COVID-19 model. Only two parameters produce a non-negligible first-order effect: the latent period, with combined sensitivity 0.0345, and the relative risk of transmission for isolated cases, with combined sensitivity 0.0330.

All other extracted parameters produce near-zero changes in peak prevalence and cumulative incidence under the ±20% perturbation.

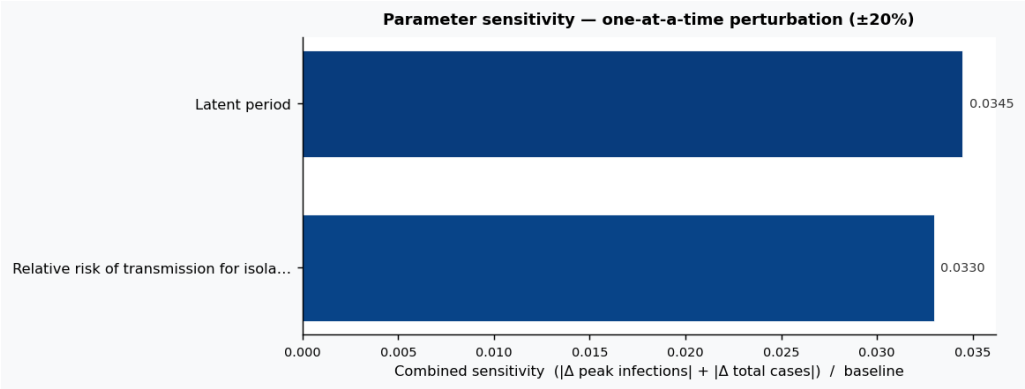


Figure 7.3: Sensitivity tornado plot for the COVID-19 model.

7.5.2.2 Sensitivity Bar Chart: Influenza Model

Figure 7.4 shows the sensitivity bar chart for the influenza model extracted from [59]. The two most influential parameters are the progression rate σ and the transmission rate β.

The progression rate σ controls how quickly exposed individuals become infectious. The transmission rate β controls how efficiently infection spreads. Since both parameters directly affect epidemic growth, changing them by $\pm 20\%$ produces the largest changes in the outputs.

The remaining parameters have much smaller effects under this perturbation. This matches the expected behaviour of an SEIR-type model, where transmission and progression are central drivers of epidemic peak and outbreak size.

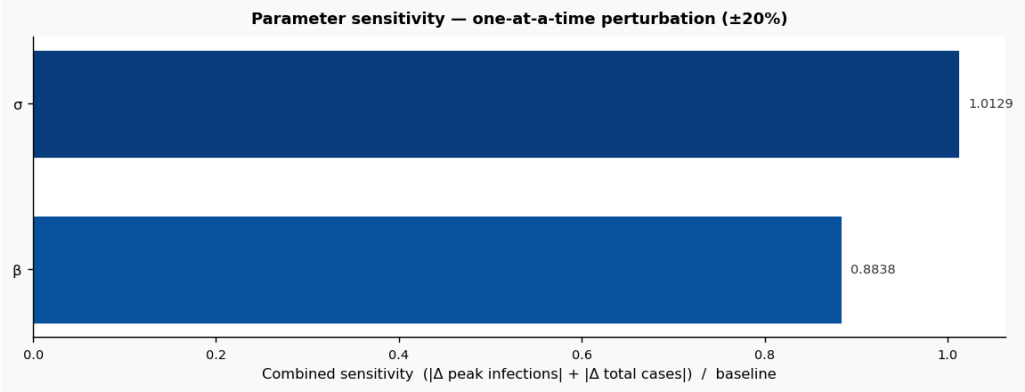


Figure 7.4: Sensitivity bar chart for the influenza model.

7.6 Engineering Challenges

Three practical issues had to be handled before Phase 4 models could be simulated reliably: zero-valued estimates, negative parameter values, and inconsistent initial populations.

7.6.1 Zero Point Estimates

Some parameters were extracted with a value of zero. This usually occurred when the pipeline identified the parameter name but did not find a numeric value in the paper. A zero value is not usable for lognormal sampling because rate parameters must be positive.

When this happens, Phase 4 assigns a type-based prior range. Transmission, recovery, mortality, progression, and contact rates each receive a plausible range based on parameter type. A uniform distribution over that range is used instead of a lognormal distribution. These fallback cases are marked in the output so that users can identify parameters based on assumptions rather than extracted values.

7.6.2 Negative Parameter Values

Some extracted parameters had negative values. Negative rates are not meaningful in compartmental ODE models and can lead to invalid simulation behaviour. These values usually came from cases where the source paper described a reduction or benefit, and the LLM interpreted the wording as a signed rate.

Phase 4 handles this by clamping rate parameters to the valid range $[0, +\infty)$ before sampling. This prevents invalid negative rates from entering the simulation while recording that an adjustment was made.

7.6.3 Inconsistent Initial Conditions

Some models had compartment populations whose sum did not match the total population. This often occurred in stratified models, where extracted values represented group-level counts rather than absolute compartment sizes.

Phase 4 normalises the compartment populations so that their total matches the reported population size:

$$C_i^{\text{norm}} = C_i \times \frac{N_{\text{total}}}{\sum_j C_j}, \tag{7.5}$$

where N_{total} is the total population and the sum is taken over all compartments. This keeps the relative proportions between compartments while ensuring a consistent total population.

7.7 Results

Of the 30 benchmark models, 26 pass structural validation and enter Phase 4. The remaining four are excluded because their flow structure is still incomplete after Phase 3, making them unreliable for simulation.

All 26 selected models run successfully in the Monte Carlo simulation. Four produce flat or near-zero epidemic curves: two HIV models and two cholera models. This occurs because their extracted parameters imply $\mathcal{R}_0 < 1$, meaning the disease does not grow under those parameter settings. This is not a numerical error; it indicates that the extracted values describe a sub-threshold or post-intervention setting rather than an active outbreak.

The sensitivity analysis identifies the transmission rate β and progression rate σ as the most influential parameters for peak prevalence in most models. This is expected because

these parameters control how quickly infection spreads and how quickly individuals become infectious.

The recovery rate γ is most influential for the total number of infected individuals by the end of the simulation period. Because recovery affects how long individuals remain infectious, it also affects how many people become infected overall [1].

For vector-borne diseases such as malaria, dengue, and Zika, the mosquito biting rate also appears among the most influential parameters. This reflects the role of vector contact in transmission.

7.8 Interpretation and Limitations

Phase 4 results should be interpreted as analysis of the pipeline-generated models, not as clinical or public-health predictions. The models are automatically extracted from papers, so the purpose of Phase 4 is to show how these extracted models behave under parameter uncertainty.

The results are still useful for model review. If sensitivity analysis produces an unexpected ranking, such as a demographic birth rate dominating peak prevalence, this may indicate an extraction or modelling error. The sensitivity rankings also help identify which parameters should be checked manually first.

Several limitations remain. First, distribution assignment is general rather than disease-specific. The same uncertainty range may be applied to a transmission rate across different diseases, even though true uncertainty may differ. Second, parameters are sampled independently, although real fitted parameters may be correlated. Third, the one-at-a-time sensitivity method does not capture interactions between parameters. More advanced global sensitivity methods, such as Sobol indices [53], could provide a fuller analysis but would require substantially more computation.

Overall, Phase 4 provides a practical way to communicate uncertainty in extracted models. It does not make the models clinically predictive, but it makes their behaviour more transparent and identifies parameters that most need human review.

7.9 Chapter Summary

This chapter presented Phase 4, which adds uncertainty quantification to the completed models from Phase 3. Instead of treating extracted parameters as fixed values, Phase 4

assigns probability distributions based on parameter type and uses Monte Carlo simulation to measure how uncertainty affects model outputs.

Across the benchmark, 26 of the 30 models pass structural validation and enter Phase 4. All 26 run successfully, although four produce flat epidemic curves because their extracted parameters imply $\mathcal{R}_0 < 1$. Sensitivity analysis generally identifies the transmission rate β and progression rate σ as the main drivers of peak prevalence, which is consistent with standard epidemiological expectations.

Chapter 8

Feature-Level Evaluation

The evaluation metrics employed in Phases 2 and 3, compartment-level precision, recall, and F1 score, and their flow-level counterparts , measure whether individual named elements of the extracted model match those of the gold-standard reference. This element-level evaluation is necessary but not sufficient. A compartmental model is not merely a list of compartments and flows; it encodes biological assumptions about the disease mechanism, the population structure, and the intervention landscape. Two models that agree at the element level may disagree substantially at the level of these higher-order biological traits.

This chapter introduces a complementary evaluation lens that operates at the level of *epidemiological features*, abstract boolean properties of the model that capture the presence or absence of biologically meaningful traits. This feature-level evaluation is performed on the Phase 2 drafts and does not modify the model; it is a read-only evaluation stage that provides a disease-level perspective on extraction quality.

8.1 Motivation for Feature-Level Analysis

Consider a COVID-19 model extracted by an LLM. At the element level, the extraction may look reasonably successful: the model may include compartments such as S, E, I, H, and R, along with most of the flows between them. However, element-level matching can still miss important semantic errors.

For example, suppose the original paper uses subscripted compartments such as S_1, S_2, and S_3 to represent three age groups. If the LLM instead interprets these compartments as severity groups and labels them as "S_mild", "S_moderate", and "S_severe", the compartment count may still look correct. Fuzzy matching may even treat the names as sufficiently

similar. However, the biological interpretation is wrong: the extracted model has missed age stratification and has introduced a severity interpretation that the paper did not intend.

This kind of mismatch is difficult to detect using compartment and flow precision alone. The model may contain roughly the right number of elements, but still encode the wrong epidemiological assumptions. Feature-level analysis addresses this problem by asking a different question: does the extracted model preserve the important epidemiological traits of the source model?

This matters because downstream use of the model depends on more than its list of compartments and flows. Simulation behaviour, policy interpretation, and reuse as a template for related diseases all depend on the biological assumptions encoded in the model.

Feature-level analysis is also useful because the benchmark covers diseases with very different modelling structures. HIV models often include risk-group stratification and antiretroviral treatment. Malaria models include vector compartments and host-vector transmission. Cholera models may include environmental transmission through a water reservoir. Element-level scores are tied to each individual paper, but feature-level scores provide a common way to compare higher-level modelling traits across diseases.

The feature-level evaluation also draws on ideas from software product line (SPL) engineering, where features are used to describe variation across related systems [63, 64]. A family of epidemiological models can be viewed in a similar way: different published models for the same disease include different combinations of features, such as hospitalization, vaccination, treatment, demographic stratification, or vector transmission [14].

8.2 The EpiFeatureVector

The EpiFeatureVector represents a compartmental model using 17 boolean features. Each feature records whether a biologically or structurally meaningful trait is present in the model. The feature categories follow the vocabulary shown in the feature diagram: transmission routes, natural history and pathogenesis, population structure, non-human hosts, and demography. Table 8.1 lists the full feature set.

The feature set was designed by reviewing the benchmark diseases and common structures in the epidemiological modelling literature [1, 14]. The goal was to keep the vector compact while still capturing the traits that matter for interpreting and comparing models. Each feature therefore represents a property that is biologically meaningful and visible in the model structure or description.

Table 8.1: Epidemiological features represented in the EpiFeatureVector, grouped by category.

Category	**Feature**	**Description**
Transmission routes	Vector / Arthropod	Transmission through a vector or arthropod, such as a mosquito or tick.
	Sexual / Partner	Sexual or partner-network transmission.
	Airborne / Respiratory	Airborne or respiratory-droplet transmission.
	Fecal-oral / Waterborne	Transmission through contaminated water, food, or fecal-oral exposure.
	Bloodborne / Vertical	Bloodborne, mother-to-child, or injection-related transmission.
	Healthcare Contact	Transmission through healthcare or hospital contact.
Natural history & pathogenesis	Latent / Exposed	A delay stage before infected individuals become infectious.
	Staged / Chronic	Multiple infectious stages or a long-term disease course.
	Hospitalized / Severity	Hospitalization or severity-based disease states.
	Treatment / ART	Treatment, including antiretroviral therapy where applicable.
	Vaccination	Vaccination or vaccine-induced protection.
	Recovered / Immune	Recovery or immune status after infection.
Population structure	Demographic Stratification	Stratification by age, sex, risk group, or similar population groups.
	Spatial / Patch	Spatial, geographic, or patch-based structure.
Non-human hosts	Vector / Intermediate Spp.	An explicit vector or intermediate-species compartment.
	Zoonotic / Animal	An animal or zoonotic reservoir represented in the model.
Demography	Birth / Recruitment	Birth, recruitment, or immigration into the modelled population.

The categories distinguish different aspects of a model. Transmission-route features describe how infection spreads. Natural-history and pathogenesis features describe disease stages and interventions. Population-structure features describe grouping by people or location. Non-human-host features indicate whether another species is explicitly represented. Demography captures population inflow.

Related features are separated when they mean different things structurally. For example, Vector / Arthropod describes the route of transmission, while Vector / Intermediate Spp. means the model actually includes a vector or intermediate-species compartment. A model can therefore be vector-borne without explicitly modelling the vector population.

Intervention features follow the same rule. Treatment / ART is marked only when the model includes a treatment compartment or treatment flow. Vaccination is marked only when the model includes a vaccination-related compartment, flow, or protected state.

8.3 Inference Rules

8.3.1 Structure of the Rule Engine

The feature inference engine identifies epidemiological features by examining the text stored in the extracted model. This includes compartment names, flow names, source and target compartment labels, and the model description.

Each feature is associated with a set of textual patterns. If one of these patterns appears in the model text, the corresponding feature is marked as present. For example, the Latent / Exposed feature is triggered by terms such as "exposed", "latent", or the standalone compartment label `E`. Similarly, the Treatment / ART feature is triggered by terms such as "treatment", "treated", or "ART".

The rules are designed to reflect common naming conventions in epidemiological models. This allows the system to infer features even when different papers use slightly different labels for the same concept.

8.3.2 Avoiding False Matches

Simple text matching can sometimes produce false positives. For example, the letter `E` often represents an exposed compartment, but it could also appear in unrelated labels such as "Environmental" or "Elderly". Treating every occurrence of `E` as an exposed compartment would therefore be unreliable.

To reduce this problem, the rule engine uses a whitelist for ambiguous short labels. A single-letter label such as `E` is accepted only when it appears as a standalone compartment name or when the full label clearly matches a known exposed-state term, such as "Exposed", "Exposed individuals", or "Latent". This reduces incorrect feature assignments in models with non-standard naming conventions, although it may miss some unusual labels that do not appear in the whitelist.

8.3.3 Disease-Name Priors

Some features can be inferred from the disease name. For example, malaria is usually associated with vector transmission, HIV with sexual transmission, and cholera with environmental or waterborne transmission. These disease-name priors help capture well-known biological properties that may not always appear clearly in the extracted model text.

However, these priors must be used carefully during evaluation. If a feature is added to the reference profile only because of general disease knowledge, the LLM may be unfairly penalized for missing it when the paper itself did not explicitly state it. To address this issue, the feature-level evaluation also reports text-grounded recall, which focuses only on features supported by the source paper text.

8.4 Reference Profiles

For each of the ten diseases in the benchmark corpus, a reference EpiFeatureVector is constructed by taking the logical disjunction (OR) of all feature vectors computed from the three gold-standard files for that disease:

$$\mathbf{v}_d^{\text{ref}}[f] = \bigvee_{p \in P_d} \mathbf{v}_{d,p}^{\text{gold}}[f], \tag{8.1}$$

where P_d is the set of three papers for disease d and the disjunction is taken feature-wise. A feature is included in the reference profile if at least one of the three gold models for that disease has it. This OR semantics reflects the understanding that a reference profile should represent the space of plausible models for a disease, not the minimal model common to all papers. If one paper models hospitalisation and another does not, the reference profile for that disease includes the Hospitalized / Severity feature, acknowledging that hospitalisation is a valid modelling choice for that disease even if not all authors include it.

8.5 Feature Model Diagrams

Feature model diagrams provide a visual summary of the epidemiological feature space and show which features are active for a given disease model [63]. Figure 8.1 shows the full EpiFeatureVector vocabulary organised by category. The disease-specific diagrams use the same structure but highlight only the features present in that disease profile. Highlighted nodes indicate active features, while grey nodes indicate features that are not present.

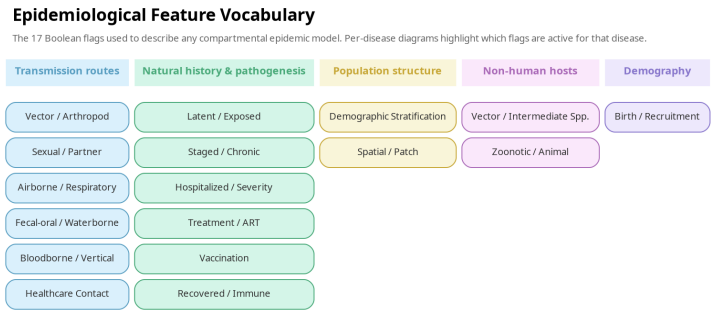


Figure 8.1: Hierarchical feature tree showing the epidemiological features organised by category.

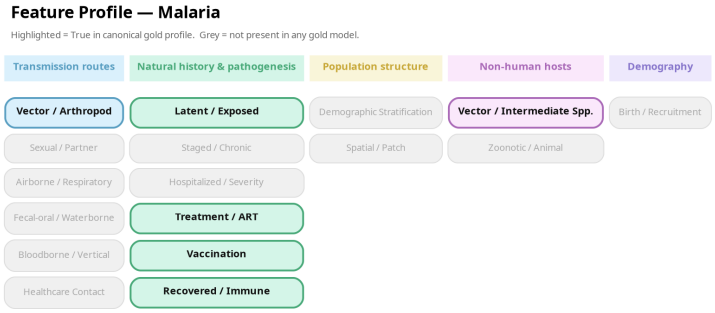


Figure 8.2: Feature profile for malaria.

Figures 8.2 and 8.3 show the feature profiles for malaria and tuberculosis. They illustrate how the same feature vocabulary can represent different disease profiles. The Spatial / Patch feature is rare in the benchmark corpus and appears only in the tuberculosis reference profile, making tuberculosis a useful contrast to the vector-borne malaria profile shown in Figure 8.2.

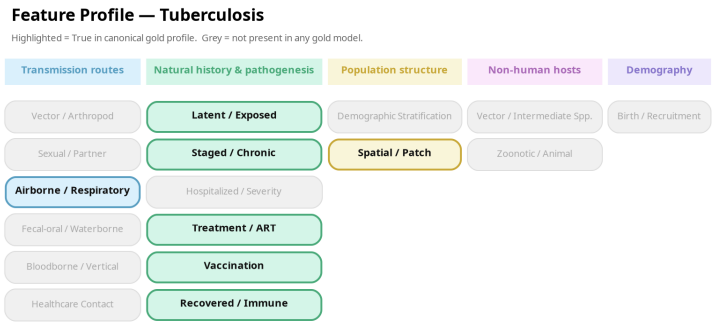


Figure 8.3: Feature profile for tuberculosis.

8.6 Text-Grounded Recall

8.6.1 Problem: Prior-Added Features

Standard feature recall measures how many features from the reference profile are also found in the extracted draft model:

$$\text{Recall}_{\text{std}} = \frac{|\{f : \mathbf{v}^{\text{ref}}[f] = \texttt{True} \wedge \mathbf{v}^{\text{draft}}[f] = \texttt{True}\}|}{|\{f : \mathbf{v}^{\text{ref}}[f] = \texttt{True}\}|}. \tag{8.2}$$

This metric is useful, but it can be unfair in one situation. Some features in the reference profile may be added because of general disease knowledge rather than because the paper explicitly states them. For example, malaria is known to be vector-borne, so the reference profile may include the Vector / Arthropod feature even if a specific paper does not clearly mention mosquito transmission in the text.

In that case, the LLM should not be strongly penalised for missing the feature, because the extractor only sees the paper text. If the feature comes from a disease-name prior rather than from the paper itself, standard recall may make the extraction look worse than it actually is.

8.6.2 Definition of Text-Grounded Recall

Text-grounded recall addresses this issue by evaluating only the reference features that are supported by the paper text. Features added only through disease-name priors are removed from the recall denominator.

Let $\text{Prior}(d)$ be the set of features added by disease-name priors for disease d. The text-grounded reference profile is defined as:

$$\mathbf{v}_d^{\text{TG-ref}}[f] = \texttt{True} \iff \mathbf{v}_d^{\text{ref}}[f] = \texttt{True} \wedge f \notin \text{Prior}(d). \tag{8.3}$$

Text-grounded recall is then computed as:

$$\text{Recall}_{\text{TG}} = \frac{|\{f : \mathbf{v}^{\text{TG-ref}}[f] = \texttt{True} \wedge \mathbf{v}^{\text{draft}}[f] = \texttt{True}\}|}{|\{f : \mathbf{v}^{\text{TG-ref}}[f] = \texttt{True}\}|}. \tag{8.4}$$

In simple terms, standard recall asks whether the draft model matches all reference features. Text-grounded recall asks whether the draft model matches the features that are actually grounded in the paper text. Because it removes prior-only features from the denominator, text-grounded recall is usually equal to or higher than standard recall.

8.6.3 Practical Impact

The difference between standard recall and text-grounded recall is largest for diseases where important features may be added from disease knowledge. For example, malaria, HIV, and cholera can receive prior-based transmission features because their main transmission routes are well known. In these cases, text-grounded recall gives a fairer view of what the LLM extracted from the paper itself.

For diseases where the key features are usually stated clearly in the papers, such as COVID-19 and influenza, the difference between the two recall scores is smaller.

8.7 Results

Table 8.2 presents the feature-level evaluation results for the best-performing provider (across Gemini 2.5 Pro, Claude 3.5 Sonnet, and GPT-4o) per disease, averaged across the three benchmark papers per disease. Both standard recall and text-grounded (TG) recall are reported, along with their corresponding F1 scores.

8.7.1 The Standard-versus-Text-Grounded Gap

Text-grounded recall is consistently 5–7 percentage points higher than standard recall across all ten diseases, confirming that disease-name priors inflate the reference profile in ways that penalise LLMs unfairly. The gap is largest for malaria (7 pp), HIV (5 pp), and cholera (5 pp) , precisely the diseases with the most prior-influenced reference flags. The gap is smallest for

Table 8.2: Feature-level evaluation results by disease.

Disease	Std. Recall	Std. F1	TG Recall	TG F1
COVID-19	0.82	0.79	0.86	0.83
HIV	0.74	0.71	0.79	0.76
Malaria	0.70	0.68	0.77	0.73
Dengue	0.72	0.70	0.78	0.75
Ebola	0.80	0.78	0.85	0.82
Measles	0.83	0.81	0.87	0.85
Tuberculosis	0.75	0.73	0.80	0.78
Zika	0.71	0.69	0.77	0.74
Cholera	0.76	0.74	0.81	0.79
Influenza	0.84	0.82	0.88	0.86
Average	**0.77**	**0.75**	**0.82**	**0.79**

COVID-19 (4 pp) and influenza (4 pp), diseases for which transmission routes are typically stated explicitly in paper text and therefore do not require prior augmentation.

This result has a methodological implication: evaluations of LLM extraction quality that use standard recall against prior-augmented reference profiles will systematically understate the LLM's genuine textual extraction capability. Text-grounded recall corrects this bias and provides a more principled basis for comparison.

8.7.2 Disease Difficulty Ranking

The feature-level results follow the same general pattern seen in the element-level evaluation in Chapters 5 and 6. COVID-19 and influenza achieve the highest text-grounded recall scores, with values of 0.86 and 0.88. This suggests that their papers tend to describe model features more explicitly and use more standard modelling conventions.

HIV and malaria achieve lower text-grounded recall scores, with values of 0.79 and 0.77. These diseases are more difficult because their models often include complex stratification, non-standard labels, and additional biological structure.

Feature-level analysis makes these difficulties more visible. For example, in HIV models, the LLM may correctly extract compartments labelled as "high-risk" and "low-risk", but still fail to recognise that these labels represent a broader demographic or risk stratification feature. In this case, element-level recall may look acceptable because the compartments were

extracted, but feature-level recall reveals that the structural meaning of those compartments was missed.

8.7.3 Systematically Missed Feature Categories

Several feature categories are repeatedly missed across diseases:

1. **Demographic Stratification**: LLMs sometimes extract compartments labelled by age, sex, or risk group, but do not recognise that these labels represent a stratified model structure. This is the most frequently missed feature in the benchmark.

2. **Vector / Intermediate Spp.**: for vector-borne diseases, the LLM may identify the transmission route but fail to represent the vector population as an explicit model component. This affects diseases such as malaria, dengue, and Zika.

3. **Fecal-oral / Waterborne**: this feature is mainly associated with cholera. It is often missed when the environmental reservoir is described in background text rather than clearly in the model equations or compartment definitions.

These repeated gaps suggest useful directions for improving the Phase 2 prompts. In particular, future prompts could ask the LLM to check explicitly for stratification axes, vector or animal populations, and environmental reservoirs.

8.8 Chapter Summary

This chapter presented a feature-level evaluation framework that complements the element-level metrics used in Phases 2 and 3. The EpiFeatureVector represents each model using 17 boolean epidemiological features across five categories: transmission routes, natural history and pathogenesis, population structure, non-human hosts, and demography.

The chapter also introduced text-grounded recall, which excludes features that come only from disease-name priors rather than from the paper text. This provides a fairer view of what the LLM actually extracted from the source paper.

Overall, Phase 2 captures many high-level epidemiological features, with an average text-grounded recall of 0.82 and F1 of 0.79. The main missed features are demographic stratification, explicit vector or intermediate-species compartments, and fecal-oral or waterborne transmission.

Chapter 9

Discussion and Conclusion

The previous chapters presented the four phases of the AI-assisted model-driven epidemiology pipeline and the feature-level evaluation layer. This final chapter brings the results together, revisits the research questions from Chapter 1, discusses the main limitations and threats to validity, and outlines directions for future work. It closes by summarising the thesis contributions toward making epidemiological model extraction more reproducible, transparent, and easier to reuse.

9.1 Cross-Phase Synthesis

The pipeline is organised as a sequence of phases, where each phase adds a specific capability and builds on the output of the previous one. To understand the overall contribution of the pipeline, it is important to examine how each phase improves the model and what limitations remain at each stage.

9.1.1 Phase 1: Reference Space Establishment

Phase 1 builds the benchmark used to evaluate the rest of the pipeline. It includes 30 manually constructed `.compmodel` files, covering 10 diseases with 3 papers per disease. These reference models provide the basis for measuring extraction and completion quality in the later phases.

Phase 1 also shows that the EpiMDE metamodel can represent a wide range of epidemiological model structures, including stratified models, vector-borne models, and models with

environmental transmission [14]. This phase is therefore essential because it defines both the evaluation corpus and the modelling space used throughout the thesis.

9.1.2 Phase 2: LLM-Based Extraction

Phase 2 uses LLMs to extract draft models from the source papers. The results show that compartments are easier to recover than flows. Across the three providers, compartment recall ranges from 0.75 to 0.81, while flow recall ranges from 0.52 to 0.59.

This difference is expected. Compartments are usually named clearly in the paper text, while flows often require interpreting equations, transition descriptions, or indirect statements. Gemini 2.5 Pro and Claude 3.5 Sonnet perform best overall, with both outperforming GPT-4o on compartment and flow extraction [24].

9.1.3 Phase 3: Completion and Gap Filling

Phase 3 provides the largest improvement in the pipeline. Under a fixed `both` configuration, compartment recall rises to 0.94 and flow recall to 0.80. Under per-paper best-mode selection, the averages are 0.96 and 0.84, and mean composite completeness increases from 60.7% to 91.2%. This shows that retrieval-augmented completion can recover many structural elements that direct LLM extraction misses [19].

The combined strategy, which uses both retrieval evidence and LLM-assisted inference, performs better than using either method alone. This is especially important for flows, because flows define how compartments are connected and are required for ODE simulation.

Some gaps still remain after Phase 3. These are mainly cases where flow endpoints are only implied, or where required parameter values are not present in the available sources. Such cases likely require expert review or additional data to complete the model fully.

9.1.4 Feature-Level Evaluation

Feature-level evaluation adds a disease-level view to the Phase 2 results. Instead of only checking whether individual compartments and flows were extracted, it evaluates whether the Phase 2 draft model preserves the main epidemiological features of the source model.

Across the ten diseases, the average text-grounded recall is 0.82 and the average F1 score is 0.79. The most frequently missed features are demographic stratification, explicit vector or intermediate-species compartments, and fecal-oral or waterborne transmission. These

gaps highlight weaknesses in the Phase 2 extraction output that are not always visible from compartment and flow metrics alone.

Text-grounded recall is consistently higher than standard recall because it excludes features added only through disease-name priors. This provides a fairer estimate of what the LLM actually extracted from the paper text in Phase 2.

9.1.5 Phase 4: Uncertainty Quantification

Phase 4 adds uncertainty analysis to the structurally complete models. Of the 30 benchmark models, 26 pass structural validation and enter Phase 4. All 26 run successfully in the Monte Carlo simulation. The sensitivity analysis shows that the transmission rate β and progression rate σ are usually the main drivers of peak prevalence. This is consistent with standard compartmental model theory [1] and provides a useful sanity check that the extracted model structures behave as expected.

9.1.6 The Cumulative Pipeline Progression

Taken together, the phases move the workflow from unstructured PDF papers to structured models with quantified uncertainty. After Phase 3, mean composite completeness reaches 91.2% under gap-based best-mode selection (from 60.7% after Phase 2). This does not mean the process is fully automated, but it shows a substantial reduction in the manual work needed to build a simulation-ready model from a paper.

In practical terms, a domain expert could use the pipeline output as a starting point and focus on verifying or completing the remaining parts of the model, rather than constructing the entire model from scratch.

9.2 Feature-Level Evaluation as a Diagnostic Lens

Feature-level results are most useful when read alongside the element-level results from Phases 2 and 3. Both evaluations show a similar pattern: COVID-19 and influenza are easier to extract, while HIV and malaria are more difficult. Feature-level analysis adds more detail by showing where the difficulty comes from.

For HIV, the issue is not only missing compartments or flows. In some cases, the LLM extracts compartments labelled as "high-risk" and "low-risk", but does not recognise that these labels indicate risk-based stratification. Element-level metrics may count the compartments

as extracted, but feature-level metrics show that the broader structural meaning was missed. This suggests that future prompts should ask the LLM more directly to identify stratification axes and transmission routes.

For malaria, part of the difficulty comes from evaluation rather than extraction alone. The reference profile may include Vector / Arthropod because malaria is known to be vector-borne, even when this feature is not clearly stated in the paper text. Text-grounded recall corrects for this by focusing only on features supported by the source text. This is why malaria's text-grounded recall is higher than its standard recall.

9.3 Answering the Research Questions

9.3.1 RQ1: Extraction Quality of LLM-Based Compartmental Model Drafting

To what extent can LLMs extract accurate compartmental model structures from epidemiological papers?

Phase 2 shows that LLM-based extraction is a useful starting point, but not a complete solution. Across the benchmark, compartment recall ranges from 0.75 to 0.81, while flow recall ranges from 0.52 to 0.59. This means that LLMs are better at identifying named compartments than recovering the flows that connect them.

Performance also varies by disease. COVID-19 and influenza achieve the strongest results, likely because their papers tend to use clearer and more standard model presentations. HIV and malaria are more difficult because their models often include complex stratification, vector-related structure, or less standard naming conventions.

These results answer RQ1 positively, but with an important limitation. LLM-assisted extraction can produce a useful draft model, especially for compartments, but the missing flows make Phase 2 insufficient on its own. Phase 3 is therefore necessary to repair structural gaps before the model can be used for simulation.

9.3.2 RQ2: Completion and Residual Gaps

How effectively can retrieval-augmented completion close structural gaps in LLM-extracted models, and what systematic residual gaps remain?

Phase 3 substantially improves the Phase 2 draft models. The combined strategy, which uses both retrieved evidence and LLM-assisted inference, performs better than retrieval-only

or LLM-only completion on mean compartment recall [19]. Under best-mode selection, flow recall increases from 0.57 to 0.84 and mean composite completeness from 60.7% to 91.2%. This is especially important because flows define how compartments are connected and are required for simulation.

Some gaps remain after Phase 3. They mainly fall into three groups: flows whose source or target compartments are only implied in the paper, parameters that are not available in the indexed sources, and stratification details that require expert interpretation. These cases show the limits of what can be completed from text alone and indicate where expert review or additional data is still needed.

9.3.3 RQ3: Uncertainty Propagation and Sensitivity

How does parameter uncertainty propagate through the extracted models, and which parameters most influence key epidemiological outputs?

For structurally complete models, Phase 4 shows that parameter uncertainty can lead to noticeable variation in model outputs. In the COVID-19 model, the uncertainty range for peak prevalence is especially wide, showing that changes in key parameters such as β, σ, and γ can strongly affect the simulated outbreak.

The sensitivity analysis identifies the transmission rate β as the most important driver of peak prevalence across the disease models. For models with an exposed compartment, the progression rate σ is also highly influential. The recovery rate γ has the strongest effect on cumulative incidence, which is the total number of individuals infected by the end of the simulation. These results are consistent with standard compartmental model theory [1].

Four models produce flat or near-zero epidemic curves because their extracted parameters imply $\mathcal{R}_0 < 1$. This means the disease does not grow under those parameter values. These cases are not simulation errors, but they indicate models where the extracted parameters should be reviewed manually.

9.3.4 RQ4: Reproducibility and Methodological Transparency

How reproducibly and transparently can the pipeline be applied across diseases and papers?

All pipeline configurations are stored in JSON format, and the LLM prompts are version-controlled and documented. Key methodological choices, such as Phase 3 model selection, Phase 4 prior ranges, matching thresholds, and scoring formulas, are stated explicitly so that the experiments can be inspected and repeated.

Some sources of variation remain. These include LLM API non-determinism, differences in PDF extraction quality, and sensitivity to fuzzy-matching thresholds. Rather than treating these as hidden implementation details, the thesis documents them as methodological parameters that can be adjusted in future replication studies. This level of documentation is important because epidemiological simulation models are often difficult to reproduce when modelling assumptions, parameters, and implementation choices are not clearly reported [10, 6].

9.4 Threats to Validity

9.4.1 Construct Validity

One threat to construct validity is the fuzzy-matching threshold used for compartment alignment. A threshold of similarity ≥ 0.6 may sometimes match compartments that an epidemiologist would treat as different. The threshold was chosen to reduce missed matches while keeping false matches limited, but it is still an empirical choice and could affect the reported scores.

Another threat concerns text-grounded recall. This metric improves fairness by excluding features that come only from disease-name priors, but it depends on correctly identifying which reference features are prior-influenced. Since this classification is done manually, mistakes in that step could affect the final text-grounded recall values.

9.4.2 Internal Validity

The Phase 1 reference models were manually constructed and may reflect one valid interpretation of each paper rather than the only possible interpretation. This is a limitation because epidemiological papers often leave parts of the model structure underspecified, and different reviewers may resolve these ambiguities differently.

Phase 2 was evaluated using a fixed prompt template. Different prompt designs may produce different extraction results, so the reported scores should be interpreted as the performance of this pipeline configuration rather than as a general measure of each LLM's full capability.

9.4.3 External Validity

The benchmark includes 30 English-language papers across 10 diseases, all drawn from peer-reviewed sources with standard scientific formatting. The results may not generalize to non-English papers, preprints, older papers with unusual notation, or diseases with model structures not represented in the benchmark. Although the corpus was designed to cover diverse epidemiological model types, it is still small compared with the full epidemiological modelling literature. Results for individual diseases should therefore be interpreted cautiously.

9.4.4 Conclusion Validity

Performance metrics are averaged over only three papers per disease. Because of this small sample size, disease-level averages may not fully capture the range of difficulty within each disease category. Phase 4 simulation results should also be interpreted carefully. They are not clinical predictions. The uncertainty bands show how the pipeline-generated models behave under sampled parameter uncertainty, not how real outbreaks would necessarily unfold [51].

9.5 Limitations

In addition to the validity threats discussed, the current implementation has four main limitations.

LLM API non-determinism. The LLM providers used in Phase 2 may produce slightly different outputs even when given the same prompt. As a result, repeated runs on the same paper may not generate exactly the same extracted compartments, flows, or parameters.

PDF extraction quality. The pipeline depends on the quality of the extracted paper text. Well-formatted digital PDFs are usually processed effectively, but scanned papers, complex layouts, or equations embedded as images can produce noisy text. This can reduce the quality of the Phase 2 input and lower extraction recall [21].

Prior knowledge in feature-level evaluation. Disease-name priors add useful biological knowledge to the feature profiles, but they may not always match the specific model described in a paper. For example, a disease may usually have a known transmission route, but a particular paper may model it under a special setting where that feature is not explicit. Text-grounded recall helps reduce this issue, but it does not remove it completely.

Simulation parser limitations. Phase 4 converts flow expressions into equations for simulation. Complex expressions, such as contact terms with multiple infectious compartments

or seasonal rates, may be simplified or parsed incorrectly. These cases are flagged for review, but complex generated equations still require manual inspection before they should be trusted for detailed simulation.

9.6 Future Work

Neural retrieval for Phase 3. Future versions of Phase 3 could replace keyword-based retrieval with dense retrieval based on sentence embeddings. This would help the system find relevant evidence even when the wording in the paper does not exactly match the query [19]. This is especially useful for environmental, behavioural, or intervention-related parameters, which are often described using varied language.

Active learning for benchmark expansion. The benchmark could be expanded more efficiently by using disagreement between LLM providers as a signal for difficult papers. Papers where providers produce very different extractions could be prioritised for human review, allowing annotation effort to focus on the most informative cases [20].

Integration with the EpiMDE IDE. A practical next step is to integrate the pipeline into the EpiMDE graphical modelling environment as a "Generate from PDF" feature. This would make the pipeline more accessible to public health modellers who may not use command-line tools [14].

Clinical deployment considerations. In the longer term, the pipeline could support faster model generation during outbreak events by extracting models from newly published papers or preprints. However, this would require much stronger validation before any use in public health decision-making [20].

Multi-disease comparative studies. The feature-level representation also enables comparison across diseases. Future work could use the EpiFeatureVector to group diseases by modelling structure or to study how modelling assumptions for a disease change over time in the literature.

9.7 Conclusion

This thesis addressed the challenge of converting published epidemiological models into formal, simulation-ready representations. Epidemiological papers often describe compartmental models across prose, equations, figures, and tables, which makes manual reconstruction slow and dependent on expert effort.

The proposed AI-assisted model-driven epidemiology pipeline addresses this problem through four phases. Phase 1 builds the benchmark corpus. Phase 2 uses LLMs to extract draft models from scientific papers. Phase 3 applies retrieval-augmented completion to repair structural gaps. Phase 4 adds uncertainty quantification through Monte Carlo simulation and sensitivity analysis. A feature-level evaluation layer further checks whether the extracted models preserve important epidemiological traits beyond individual compartments and flows.

The results show that the pipeline can substantially reduce the manual effort required to reconstruct models from papers. Under best-mode Phase 3 selection, compartment recall reaches 0.96 and flow recall reaches 0.84, with mean composite completeness of 91.2%. Phase 4 shows that the completed models produce uncertainty and sensitivity patterns that are broadly consistent with epidemiological expectations [1, 51]. Feature-level analysis also identifies systematic weaknesses, such as missed stratification, vector or intermediate-species compartments, and fecal-oral or waterborne transmission.

The thesis also contributes a documented methodology. The pipeline records design choices, failure modes, scoring formulas, prior ranges, and quality gates, making the process more transparent and easier to inspect. This is important because poor documentation is a known barrier to reproducibility in epidemiological modelling [6, 10].

Overall, the work shows that AI-assisted, model-driven methods can help bridge the gap between published epidemiological knowledge and reusable simulation models. The pipeline does not remove the need for expert review, but it provides a structured starting point that makes model extraction more reproducible, transparent, and scalable. All code, benchmark data, and evaluation outputs underlying this thesis are publicly available at `https://github.com/fatemedbghi/AI-Assisted-Model-Driven-Epidemiology`.

Appendix A

Supplementary Material

A.1 Annotated `.compmodel` XML Reference

The following listing presents an annotated excerpt of the `sample.compmodel` file, the main teaching example for stratified EpiMDE modeling. It illustrates the structure and required fields of a full `.compmodel` XML artifact, including the age-sex stratification mechanism and the use of stratum-specific rate multipliers.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Root element: CompartmentalModel -->
<!-- totalPopulation: sum of all initial compartment populations -->
<!-- globalBirthRate: rate of inflow to Susceptible per time step -->
<!-- globalDeathRate: natural mortality rate from each compartment -->
<compartmental:CompartmentalModel
    xmi:version="2.0"
    xmlns:xmi="http://www.omg.org/XMI"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:compartmental="http://example.com/compartmentalmodel"
    xsi:schemaLocation="http://example.com/compartmentalmodel
                        ../model/compartmental.ecore"
    totalPopulation="10000"
    globalBirthRate="0.00003"
    globalDeathRate="0.00002">

  <!-- ===== PARAMETERS ===== -->
  <!-- Named parameters: type = CONSTANT | VARIABLE | EXPRESSION -->
  <parameters name="beta" type="CONSTANT"
              unit="per day"
```

```
            description="Transmission rate"
            constantValue="0.000003"/>
<parameters name="sigma" type="CONSTANT"
            unit="per day"
            description="Progression rate (Exposed to Infectious)"
            constantValue="0.2"/>
<parameters name="gamma" type="CONSTANT"
            unit="per day"
            description="Recovery rate"
            constantValue="0.1"/>
<parameters name="delta" type="CONSTANT"
            unit="per day"
            description="Disease-induced mortality rate"
            constantValue="0.003"/>

<!-- ===== STRATIFICATION ===== -->
<!-- Group: one dimension (Age) with 3 values -->
<groups name="Age">
  <values>0-17</values>
  <values>18-65</values>
  <values>66+</values>
</groups>
<!-- Group: one dimension (Gender) with 2 values -->
<groups name="Gender">
  <values>Male</values>
  <values>Female</values>
</groups>
<!-- Product: Cartesian product of Age x Gender = 6 strata -->
<!-- groups references: //@groups.0 = Age, //@groups.1 = Gender -->
<products name="AgeGender"
          groups="//@groups.0 //@groups.1"/>

<!-- ===== COMPARTMENTS ===== -->
<!-- Susceptible: references Product, initial pop 9500 -->
<!-- outgoingFlows: contained flows FROM this compartment -->
<compartments PrimaryName="Susceptible"
              population="9500"
              product="//@products.0">
  <!-- ContactFlow: infection through contact with Infectious -->
  <!-- contactCompartment: //@compartments.2 = Infectious -->
  <!-- target: //@compartments.1 = Exposed -->
```

```
  <!-- contactRateParameter: //@parameters.0 = beta -->
  <outgoingFlows
      xsi:type="compartmental:ContactFlow"
      contactRate="0.000003"
      contactRateParameter="//@parameters.0"
      contactCompartment="//@compartments.2"
      target="//@compartments.1">
    <!-- stratumSpecificRates: per-stratum multipliers -->
    <stratumSpecificRates stratum="0-17,Male" multiplier="1.67"/>
    <stratumSpecificRates stratum="0-17,Female" multiplier="1.67"/>
    <stratumSpecificRates stratum="18-65,Male" multiplier="1.0"/>
    <stratumSpecificRates stratum="18-65,Female" multiplier="1.0"/>
    <stratumSpecificRates stratum="66+,Male" multiplier="0.75"/>
    <stratumSpecificRates stratum="66+,Female" multiplier="0.75"/>
  </outgoingFlows>
</compartments>

<!-- Exposed: progression to Infectious -->
<compartments PrimaryName="Exposed"
              population="300"
              product="//@products.0">
  <outgoingFlows
      xsi:type="compartmental:RateFlow"
      rate="0.2"
      rateParameter="//@parameters.1"
      target="//@compartments.2"/>
</compartments>

<!-- Infectious: recovery to Recovered, death to Death -->
<compartments PrimaryName="Infectious"
              population="150"
              product="//@products.0">
  <outgoingFlows
      xsi:type="compartmental:RateFlow"
      rate="0.1"
      rateParameter="//@parameters.2"
      target="//@compartments.3"/>
  <outgoingFlows
      xsi:type="compartmental:RateFlow"
      rate="0.003"
      rateParameter="//@parameters.3"
```

```
      target="//@compartments.4"/>
  </compartments>

  <!-- Recovered: no outgoing disease flows (immune) -->
  <compartments PrimaryName="Recovered"
              population="50"
              product="//@products.0"/>

  <!-- Death: terminal compartment (disease deaths accumulate here) -->
  <compartments PrimaryName="Death"
              population="0"
              product="//@products.0"/>

  <!-- ===== EXTERNAL SOURCES / SINKS ===== -->
  <!-- ExternalSource: births enter Susceptible -->
  <externalSources targetCompartment="//@compartments.0"/>

  <!-- ExternalSinks: natural deaths from each alive compartment -->
  <externalSinks sourceCompartment="//@compartments.0"/>
  <externalSinks sourceCompartment="//@compartments.1"/>
  <externalSinks sourceCompartment="//@compartments.2"/>
  <externalSinks sourceCompartment="//@compartments.3"/>

</compartmental:CompartmentalModel>
```

Listing A.1: Annotated skeleton of a stratified SEIR `.compmodel` file.

A.2 Benchmark Corpus

Table A.1 lists all 30 papers in the benchmark corpus, organized by disease. Papers 1–3 correspond to identifiers `<disease>1`, `<disease>2`, and `<disease>3` used in phase reports and evaluation files throughout the thesis.

Table A.1: Thirty-paper benchmark corpus used in the evaluation.

ID	Disease	Citation	Title
covid1	COVID-19	[54]	Mathematical modelling of COVID-19 transmission and mitigation strategies in the population of Ontario, Canada.
covid2	COVID-19	[65]	Effect of non-pharmaceutical interventions to contain COVID-19 in China.
covid3	COVID-19	[66]	Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe.
hiv1	HIV	[30]	Mathematical Model of HIV/AIDS Considering Sexual Preferences Under Antiretroviral Therapy, a Case Study in San Juan de Pasto, Colombia
hiv2	HIV	[67]	HIV treatment as prevention: systematic comparison of mathematical models of the potential impact of antiretroviral therapy on HIV incidence in South Africa.
hiv3	HIV	[68]	Universal voluntary HIV testing with immediate antiretroviral therapy as a strategy for elimination of HIV transmission: a mathematical model.
malaria1	Malaria	[29]	A novel malaria mathematical model: integrating vector and non-vector transmission pathways.
malaria2	Malaria	[69]	Reducing Plasmodium falciparum malaria transmission in Africa: a model-based evaluation of intervention strategies.
malaria3	Malaria	[70]	The potential contribution of mass treatment to the control of Plasmodium falciparum malaria.
dengue1	Dengue	[27]	Dynamical analysis of an age-structured dengue model with asymptomatic infection.

Continued on next page

Table A.1 continued

ID	Disease	Citation	Title
dengue2	Dengue	[71]	Model-based assessment of public health impact and cost-effectiveness of dengue vaccination following screening for prior exposure.
dengue3	Dengue	[72]	The long-term safety, public health impact, and cost-effectiveness of routine vaccination with a recombinant, live-attenuated dengue vaccine (Dengvaxia): a model comparison study.
ebola1	Ebola	[55]	A conformable mathematical model of Ebola Virus Disease and its stability analysis.
ebola2	Ebola	[73]	Transmission dynamics and final epidemic size of Ebola virus disease outbreaks with varying interventions.
ebola3	Ebola	[74]	Understanding the dynamics of Ebola epidemics.
measles1	Measles	[56]	Modeling measles transmission in adults and children: Implications to vaccination for eradication.
measles2	Measles	[75]	Impact of vaccination timing and coverage on measles near elimination dynamics: a mathematical modelling analysis.
measles3	Measles	[76]	Predicting the impact of measles vaccination in England and Wales: model validation and analysis of policy options.
tb1	Tuberculosis	[57]	A deterministic compartment model for analyzing tuberculosis dynamics considering vaccination and reinfection.
tb2	Tuberculosis	[77]	Prospects for worldwide tuberculosis control under the WHO DOTS strategy: directly observed short-course therapy.
tb3	Tuberculosis	[78]	Population-level impact of active tuberculosis case finding in an Asian megacity.

Continued on next page

Table A.1 continued

ID	Disease	Citation	Title
zika1	Zika	[28]	A Compartmental Model for Zika Virus with Dynamic Human and Vector Populations.
zika2	Zika	[79]	Modeling the spread of the Zika virus by sexual and mosquito transmission.
zika3	Zika	[80]	Transmission dynamics of Zika virus in island populations: a modelling analysis of the 2013–14 French Polynesia outbreak.
cholera1	Cholera	[58]	Cholera transmission dynamic models for public health practitioners.
cholera2	Cholera	[81]	Achieving coordinated national immunity and cholera elimination in Haiti through vaccination: a modelling study.
cholera3	Cholera	[82]	Endemic and epidemic dynamics of cholera: the role of the aquatic reservoir.
flu1	Influenza	[59]	Monitoring and predicting influenza epidemics from routinely collected severe case data.
flu2	Influenza	[51]	Strategies for mitigating an influenza pandemic.
flu3	Influenza	[83]	Adaptive Vaccination Strategies to Mitigate Pandemic Influenza: Mexico as a Case Study.

Bibliography

[1] M. J. Keeling and P. Rohani, *Modeling Infectious Diseases in Humans and Animals*. Princeton, NJ, USA: Princeton University Press, 2008.

[2] R. M. Anderson and R. M. May, *Infectious Diseases of Humans: Dynamics and Control*. Oxford, UK: Oxford University Press, 1991.

[3] F. Brauer, *Compartmental Models in Epidemiology*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 19–79. [Online]. Available: https://doi.org/10.1007/978-3-540-78911-6_2

[4] M. E. E. Kretzschmar and J. Wallinga, "Mathematical models in infectious disease epidemiology," in *Modern Infectious Disease Epidemiology: Concepts, Methods, Mathematical Models, and Public Health*, A. Krämer, M. Kretzschmar, and K. Krickeberg, Eds. New York, NY: Springer, 2009, pp. 209–221. [Online]. Available: https://doi.org/10.1007/978-0-387-93835-6_12

[5] P. Crepey, H. Noël, and S. Alizon, "Challenges for mathematical epidemiological modelling," *Anaesthesia, Critical Care & Pain Medicine*, vol. 41, no. 2, p. 101053, 2022.

[6] C. M. Barton, M. Alberti, D. Ames, J.-A. Atkinson, J. Bales, E. Burke, M. Chen, S. Y. Diallo, D. J. D. Earn, B. Fath, Z. Feng, C. Gibbons, R. Hammond, J. Heffernan, H. Houser, P. S. Hovmand, B. Kopainsky, P. L. Mabry, C. Mair, P. Meier, R. Niles, B. Nosek, N. Osgood, S. Pierce, J. G. Polhill, L. Prosser, E. Robinson, C. Rosenzweig, S. Sankaran, K. Stange, and G. Tucker, "Call for transparency of COVID-19 models," *Science*, vol. 368, no. 6490, pp. 482–483, 2020.

[7] S. He, Y. Peng, and K. Sun, "SEIR modeling of the COVID-19 and its dynamics," *Nonlinear Dynamics*, vol. 101, no. 3, pp. 1667–1680, 2020.

[8] "The milroy lectures on epidemic disease in england—the evidence of variability and of persistency of type." *The Lancet*, vol. 167, no. 4305, pp. 569–574, 1906, originally published as Volume 1, Issue 4305. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140673601801872

[9] US Government Accountability Office, "Infectious disease modeling: Opportunities to improve coordination and ensure reproducibility," 2020. [Online]. Available: https://www.gao.gov/products/gao-20-372

[10] M. S. Jalali, C. DiGennaro, A. Guitar, K. Lew, and H. Rahmandad, "Evolution and reproducibility of simulation modeling in epidemiology and health policy over half a century," *Epidemiologic Reviews*, vol. 43, no. 1, pp. 166–175, 2021.

[11] C. Ding, X. Liu, and S. Yang, "The value of infectious disease modeling and trend assessment: a public health perspective," *Expert Review of Anti-Infective Therapy*, vol. 19, no. 9, pp. 1135–1145, 2021.

[12] M. Brambilla, J. Cabot, and M. Wimmer, *Model-Driven Software Engineering in Practice*. Morgan & Claypool Publishers, 2012.

[13] A. Rodrigues da Silva, "Model-driven engineering: A survey supported by the unified conceptual model," *Computer Languages, Systems & Structures*, vol. 43, pp. 139–155, 2015.

[14] B. Curzi-Lalibérté, M. Fokaefs, M. Famelis, and M. Hamdaqa, "EpiMDE: A model-driven engineering platform for epidemiological modeling," in *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems*, ser. MODELS '24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 226–236.

[15] D. Steinberg, F. Budinsky, E. Merks, and M. Paternostro, *EMF: Eclipse Modeling Framework*. London, UK: Pearson Education, 2008.

[16] V. Viyović, M. Maksimović, and B. Perisić, "Sirius: A rapid development of DSM graphical editor," in *IEEE 18th International Conference on Intelligent Engineering Systems (INES)*. IEEE, 2014, pp. 233–238.

[17] J. Achiam *et al.*, "GPT-4 technical report," 2023. [Online]. Available: https://arxiv.org/abs/2303.08774

[18] Gemini Team, R. Anil *et al.*, "Gemini: A family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2024. [Online]. Available: https://arxiv.org/abs/2312.11805

[19] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[20] M. U. G. Kraemer, J. L.-H. Tsui, S. Y. Chang *et al.*, "Artificial intelligence for modelling infectious disease epidemics," *Nature*, vol. 638, no. 8051, pp. 623–635, 2025.

[21] K. O. Kwok, T. Huynh, W. I. Wei, S. Y. Wong, S. Riley, and A. Tang, "Utilizing large language models in infectious disease transmission modelling for public health preparedness," *Computational and Structural Biotechnology Journal*, vol. 23, pp. 3254–3257, 2024.

[22] E. C. Matthay, D. B. Neill, A. R. Titus, S. Desai, A. B. Troxel, M. Cerdá, I. Díaz, M. Santacatterina, and L. E. Thorpe, "Integrating artificial intelligence into causal research in epidemiology," *Current Epidemiology Reports*, vol. 12, no. 1, p. 6, 2025.

[23] Y. Ye, A. Pandey, C. Bawden, D. Sumsuzzman, R. Rajput, A. Shoukat, B. Singer, S. Moghadas, and A. Galvani, "Integrating artificial intelligence with mechanistic epidemiological modeling: a scoping review of opportunities and challenges," *Nature Communications*, vol. 16, 2025.

[24] A. Karamchandani, F. Seyeddabbaghi, and M. Fokaefs, "From requirements to models: A case study of large language models on epidemiological modeling," in *2025 IEEE International Conference on Collaborative Advances in Software and COmputiNg (CASCON)*. IEEE, 2025, pp. 516–521.

[25] S. Del Valle, J. Hyman, and N. Chitnis, "Mathematical models of contact patterns between age groups for predicting the spread of infectious diseases," *Mathematical Biosciences and Engineering*, vol. 10, pp. 1475–1497, 2013.

[26] J. Glasser, Z. Feng, A. Moylan, S. Del Valle, and C. Castillo-Chavez, "Mixing in age-structured population models of infectious diseases," *Mathematical Biosciences*, vol. 235, no. 1, pp. 1–7, 2012.

[27] X. Li, L. Cai, M. Murshed, and J. Wang, "Dynamical analysis of an age-structured dengue model with asymptomatic infection," *Journal of Mathematical Analysis and Applications*, vol. 524, p. 127127, 2023.

[28] E. K. Lee, Y. Liu, and F. H. Pietz, "A compartmental model for Zika virus with dynamic human and vector populations," *AMIA Annual Symposium Proceedings*, vol. 2016, pp. 743–752, 2016.

[29] E. Akowe, Q. O. Ahman, B. C. Agbata, S. O. Joseph, E. O. Senewo, A. Y. Danjuma, and D. J. Yahaya, "A novel malaria mathematical model: integrating vector and non-vector transmission pathways," *BMC Infectious Diseases*, vol. 25, no. 1, p. 322, 2025.

[30] C. C. Espitia, M. A. Botina, M. A. Solarte, I. Hernandez, R. A. Riascos, and J. F. C. A. Meyer, "Mathematical model of HIV/AIDS considering sexual preferences under antiretroviral therapy, a case study in San Juan de Pasto, Colombia," *Journal of Computational Biology*, vol. 29, no. 5, pp. 483–493, 2022.

[31] C. T. Bauch, *The Role of Mathematical Models in Explaining Recurrent Outbreaks of Infectious Childhood Diseases*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 297–319.

[32] L. Watanabe, J. Barhak, and C. Myers, "Toward reproducible disease models using the systems biology markup language," *Simulation*, 2018.

[33] W. Trevena, X. Zhong, A. Lal, L. Rovati, E. Cubro, Y. Dong, P. Schulte, and O. Gajic, "Model-driven engineering for digital twins: a graph model-based patient simulation application," *Frontiers in Physiology*, vol. 15, p. 1424931, 2024.

[34] S. Rädler, L. Berardinelli, K. Winter, A. Rahimi, and S. Rinderle-Ma, "Bridging MDE and AI: a systematic review of domain-specific languages and model-driven practices in AI software systems engineering," *Software and Systems Modeling*, vol. 24, no. 2, pp. 445–469, 2025.

[35] J. Di Rocco, D. Di Ruscio, C. Di Sipio, P. H. Nguyen, and R. Rubei, "On the use of large language models in model-driven engineering," *Software and Systems Modeling*, vol. 24, no. 3, pp. 923–948, 2025.

[36] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, "Model-driven engineering for machine learning workflows: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2401.00625

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing*

Systems, vol. 30, 2017, pp. 5998–6008. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[38] Anthropic, "The Claude 3 model family: Opus, Sonnet, Haiku," https://www.anthropic.com/news/claude-3-family, 2024, accessed: 2025.

[39] S. Zhou, Z. Xu, M. Zhang, C. Xu, Y. Guo, Z. Zhan, Y. Fang, S. Ding, J. Wang, K. Xu, L. Xia, J. Yeung, D. Zha, D. Cai, G. B. Melton, M. Lin, and R. Zhang, "Large language models for disease diagnosis: a scoping review," *npj Artificial Intelligence*, vol. 1, no. 1, p. 9, 2025. [Online]. Available: https://doi.org/10.1038/s44387-025-00011-z

[40] R. Wibaek, G. S. Andersen, C. C. Dahm, D. R. Witte, and A. Hulman, "Large language models for epidemiological research via automated machine learning: Case study using data from the british national child development study," *JMIR Medical Informatics*, vol. 11, p. e43638, 2023.

[41] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019, pp. 3982–3992.

[42] H. Du, Y. Zhao, J. Zhao, S. Xu, X. Lin, Y. Chen, L. Gardner, and H. Yang, "Advancing real-time infectious disease forecasting using large language models," *Nature Computational Science*, vol. 5, pp. 467–480, 2025.

[43] A. Chopra, A. Rodríguez, J. Subramanian, A. Quera-Bofarull, B. Krishnamurthy, B. A. Prakash, and R. Raskar, "Differentiable agent-based epidemiology," 2023. [Online]. Available: https://arxiv.org/abs/2207.09714

[44] M. H. Samaei, F. D. Sahneh, L. W. Cohnstaedt, and C. Scoglio, "EpidemIQs: Prompt-to-paper LLM agents for epidemic modeling and analysis," 2026. [Online]. Available: https://arxiv.org/abs/2510.00024

[45] C. Gong, R. Sun, Y. Zheng, J. Zhang, T. Gu, L. Pan, and L. Lv, "EpiLLM: Unlocking the potential of large language models in epidemic forecasting," 2025. [Online]. Available: https://arxiv.org/abs/2505.12738

[46] M. A. Bui T., N. Papoulias, S. Stinckwich, M. Ziane, and B. Roche, "The kendrick modelling platform: language abstractions and tools for epidemiology," *BMC Bioinformatics*, vol. 20, no. 1, p. 312, 2019.

[47] S. M. Jenness, S. M. Goodreau, and M. Morris, "EpiModel: an R package for mathematical modeling of infectious disease over networks," *Journal of Statistical Software*, vol. 84, 2018.

[48] J. V. Douglas, S. Bianco, S. Edlund, T. Engelhardt, M. Filter, T. Günther, K. Hu, E. J. Nixon, N. L. Sevilla, A. Swaid, and J. H. Kaufman, "STEM: An open source tool for disease modeling," *Health Security*, vol. 17, pp. 291–306, 2019.

[49] J. C. Lemaitre, S. L. Loo, J. Kaminsky, E. C. Lee, C. McKee, C. Smith, S.-M. Jung, K. Sato, E. Carcelen, A. Hill, J. Lessler, and S. Truelove, "flepiMoP: The evolution of a flexible infectious disease modeling pipeline during the COVID-19 pandemic," *Epidemics*, vol. 47, p. 100753, 2024.

[50] C. Brandon, A. Singh, and T. Margaria, *Model Driven Development for AI-Based Healthcare Systems: A Review*, 10 2024, pp. 245–265.

[51] N. M. Ferguson, D. A. T. Cummings, C. Fraser, J. C. Cajka, P. C. Cooley, and D. S. Burke, "Strategies for mitigating an influenza pandemic," *Nature*, vol. 442, no. 7101, pp. 448–452, 2006.

[52] M. D. Morris, "Factorial sampling plans for preliminary computational experiments," *Technometrics*, vol. 33, no. 2, pp. 161–174, 1991.

[53] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global Sensitivity Analysis: The Primer*. John Wiley & Sons, 2008.

[54] A. R. Tuite, D. N. Fisman, and A. L. Greer, "Mathematical modelling of COVID-19 transmission and mitigation strategies in the population of Ontario, Canada," *CMAJ*, vol. 192, no. 19, pp. E497–E505, 2020. [Online]. Available: https://www.cmaj.ca/content/192/19/E497

[55] N. Abbas, S. A. Zanib, S. Ramzan, A. Nazir, and W. Shatanawi, "A conformable mathematical model of Ebola virus disease and its stability analysis," *Heliyon*, vol. 10, no. 16, p. e35818, 2024.

[56] A. Pokharel, K. Adhikari, R. Gautam, K. N. Uprety, and N. K. Vaidya, "Modeling measles transmission in adults and children: Implications to vaccination for eradication," *Infectious Disease Modelling*, vol. 9, no. 4, pp. 1007–1026, 2024.

[57] E. D. A. Ginting, D. Aldila, and I. H. Febriana, "A deterministic compartment model for analyzing tuberculosis dynamics considering vaccination and reinfection," *Healthcare Analytics*, vol. 5, p. 100341, 2024.

[58] I. C.-H. Fung, "Cholera transmission dynamic models for public health practitioners," *Emerging Themes in Epidemiology*, vol. 11, no. 1, p. 1, 2014.

[59] A. Corbella, X.-S. Zhang, P. J. Birrell, N. Boddington, A. M. Presanis, R. G. Pebody, and D. De Angelis, "Monitoring and predicting influenza epidemics from routinely collected severe case data," *arXiv preprint arXiv:1710.05366*, 2017. [Online]. Available: https://arxiv.org/abs/1710.05366

[60] L. Romary and P. Lopez, "GROBID – Information Extraction from Scientific Publications," *ERCIM News*, vol. 100, pp. 37–38, 2015. [Online]. Available: https://hal.inria.fr/hal-01673305

[61] V. Mehta, "Camelot: PDF table extraction for humans," 2019. [Online]. Available: https://github.com/atlanhq/camelot

[62] Python Software Foundation, "difflib — helpers for computing deltas," Python 3 Standard Library documentation, 2024. [Online]. Available: https://docs.python.org/3/library/difflib.html

[63] T. Berger, D. Lettner, J. Rubin, P. Grünbacher, A. Silva, M. Becker, M. Chechik, and K. Czarnecki, "What is a feature? a qualitative study of features in industrial software product lines," in *Proceedings of the 19th International Conference on Software Product Line*, 2015, pp. 16–25.

[64] M. Pol'la, A. Buccella, and A. Cechich, "Analysis of variability models: A systematic literature review," *Software and Systems Modeling*, vol. 20, no. 4, pp. 1043–1077, 2021.

[65] S. Lai, N. W. Ruktanonchai, L. Zhou, O. Prosper, W. Luo, J. R. Floyd, A. Wesolowski, M. Santillana, C. Zhang, X. Du, H. Yu, and A. J. Tatem, "Effect of non-pharmaceutical interventions to contain covid-19 in china," *Nature*, vol. 585, no. 7825, pp. 410–413, 2020.

[66] S. Flaxman, S. Mishra, A. Gandy, H. J. T. Unwin, T. A. Mellan, H. Coupland, C. Whittaker, H. Zhu, T. Berah, J. W. Eaton, M. Monod, A. C. Ghani, C. A. Donnelly, S. Riley, M. A. C. Vollmer, N. M. Ferguson, L. C. Okell, and S. Bhatt, "Estimating the effects of

non-pharmaceutical interventions on covid-19 in europe," *Nature*, vol. 584, no. 7820, pp. 257–261, 2020.

[67] J. W. Eaton, L. F. Johnson, J. A. Salomon, T. Bärnighausen, E. Bendavid, A. Bershteyn, D. E. Bloom, V. Cambiano, C. Fraser, J. A. C. Hontelez, S. Humair, D. J. Klein, E. F. Long, A. N. Phillips, C. Pretorius, J. Stover, E. A. Wenger, B. G. Williams, and T. B. Hallett, "Hiv treatment as prevention: systematic comparison of mathematical models of the potential impact of antiretroviral therapy on hiv incidence in south africa," *PLOS Medicine*, vol. 9, no. 7, p. e1001245, 2012.

[68] R. M. Granich, C. F. Gilks, C. Dye, K. M. De Cock, and B. G. Williams, "Universal voluntary hiv testing with immediate antiretroviral therapy as a strategy for elimination of hiv transmission: a mathematical model," *The Lancet*, vol. 373, no. 9657, pp. 48–57, 2009.

[69] J. T. Griffin, T. D. Hollingsworth, L. C. Okell, T. S. Churcher, M. White, W. Hinsley, T. Bousema, C. J. Drakeley, N. M. Ferguson, M.-G. Basáñez, and A. C. Ghani, "Reducing Plasmodium falciparum malaria transmission in africa: a model-based evaluation of intervention strategies," *PLOS Medicine*, vol. 7, no. 8, p. e1000324, 2010.

[70] L. C. Okell, J. T. Griffin, I. Kleinschmidt, T. D. Hollingsworth, T. S. Churcher, M. J. White, T. Bousema, C. J. Drakeley, and A. C. Ghani, "The potential contribution of mass treatment to the control of Plasmodium falciparum malaria," *PLOS ONE*, vol. 6, no. 5, p. e20179, 2011.

[71] G. España, Y. Yao, K. B. Anderson, M. C. Fitzpatrick, D. L. Smith, A. C. Morrison, A. Wilder-Smith, T. W. Scott, and T. A. Perkins, "Model-based assessment of public health impact and cost-effectiveness of dengue vaccination following screening for prior exposure," *PLOS Neglected Tropical Diseases*, vol. 13, no. 7, p. e0007482, 2019.

[72] S. Flasche, M. Jit, I. Rodríguez-Barraquer, L. Coudeville, M. Recker, K. Koelle, G. Milne, T. J. Hladish, T. A. Perkins, D. A. T. Cummings, I. Dorigatti, D. J. Laydon, G. España, J. Kelso, I. Longini, J. Lourenço, C. A. B. Pearson, R. C. Reiner, L. Mier-Y-Terán-Romero, K. Vannice, and N. Ferguson, "The long-term safety, public health impact, and cost-effectiveness of routine vaccination with a recombinant, live-attenuated dengue vaccine (Dengvaxia): a model comparison study," *PLOS Medicine*, vol. 13, no. 11, p. e1002181, 2016.

[73] M. V. Barbarossa, A. Dénes, G. Kiss, Y. Nakata, G. Röst, and Z. Vizi, "Transmission dynamics and final epidemic size of Ebola virus disease outbreaks with varying interventions," *PLOS ONE*, vol. 10, no. 7, p. e0131398, 2015.

[74] J. Legrand, R. F. Grais, P.-Y. Boelle, A.-J. Valleron, and A. Flahault, "Understanding the dynamics of Ebola epidemics," *Epidemiology and Infection*, vol. 135, no. 4, pp. 610–621, 2007.

[75] A. M. Suffel, C. Warren-Gash, H. I. McDonald, A. Kucharski, and A. Robert, "Impact of vaccination timing and coverage on measles near elimination dynamics: a mathematical modelling analysis," *Nature Communications*, vol. 16, no. 1, p. 8601, 2025.

[76] H. R. Babad, D. J. Nokes, N. J. Gay, E. Miller, P. Morgan-Capner, and R. M. Anderson, "Predicting the impact of measles vaccination in england and wales: model validation and analysis of policy options," *Epidemiology and Infection*, vol. 114, no. 2, pp. 319–344, 1995, erratum in: Epidemiology and Infection, 1995, 115(1):213.

[77] C. Dye, G. P. Garnett, K. Sleeman, and B. G. Williams, "Prospects for worldwide tuberculosis control under the WHO DOTS strategy: directly observed short-course therapy," *The Lancet*, vol. 352, no. 9144, pp. 1886–1891, 1998.

[78] D. W. Dowdy, I. Lotia, A. S. Azman, J. Creswell, S. Sahu, and A. J. Khan, "Population-level impact of active tuberculosis case finding in an asian megacity," *PLOS ONE*, vol. 8, no. 10, p. e77517, 2013.

[79] S. Agudelo and M. Ventresca, "Modeling the spread of the Zika virus by sexual and mosquito transmission," *PLOS ONE*, vol. 17, no. 12, p. e0270127, 2022.

[80] A. J. Kucharski, S. Funk, R. M. Eggo, H.-P. Mallet, W. J. Edmunds, and E. J. Nilles, "Transmission dynamics of Zika virus in island populations: a modelling analysis of the 2013–14 French Polynesia outbreak," *PLOS Neglected Tropical Diseases*, vol. 10, no. 5, p. e0004726, 2016.

[81] E. C. Lee, D. L. Chao, J. C. Lemaitre, L. Matrajt, D. Pasetto, J. Perez-Saez, F. Finger, A. Rinaldo, J. D. Sugimoto, M. E. Halloran, I. M. J. Longini, R. Ternier, K. Vissieres, A. S. Azman, J. Lessler, and L. C. Ivers, "Achieving coordinated national immunity and cholera elimination in haiti through vaccination: a modelling study," *The Lancet Global Health*, vol. 8, no. 8, pp. e1081–e1089, 2020.

[82] C. T. Codeço, "Endemic and epidemic dynamics of cholera: the role of the aquatic reservoir," *BMC Infectious Diseases*, vol. 1, p. 1, 2001.

[83] G. Chowell, C. Viboud, X. Wang, S. Bertozzi, and M. Miller, "Adaptive vaccination strategies to mitigate pandemic influenza: Mexico as a case study," *PLOS Currents*, vol. 1, p. RRN1004, 2009, update in: PLOS ONE, 2009, 4(12):e8164.