BIOAUTOCON: AI for automatic model construction from Biological Networks

Submitted by:

Yash Goyal(2022588), Devansh Kumar(2022152)
B.Tech Project Report
BTP Advisor: Sriram K

Date: December 2, 2025

Indraprastha Institute of Information Technology, New Delhi

Student's Declaration

I hereby declare that the work presented in the report entitled **BioAutoCon** submitted by me for the partial fulfillment of the requirements for the degree of *Bachelor of Technology* in *Computer Science & Applied Mathematics* at Indraprastha Institute of Information Technology, Delhi, is an authentic record of my work carried out under guidance of **Dr. Sriramk K**. Due acknowledgements have been given in the report to all material used. This work has not been submitted anywhere else for the reward of any other degree.

.... **Date: December 2, 2025**
Yash Goyal, Devansh Kumar

Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

.... **Place & Date: ....**
..(advisors' name)...

Abstract

This report presents a multimodal artificial intelligence framework designed for the automated interpretation of biological network diagrams. The system processes both visual and textual inputs to extract structured biochemical information, including reaction networks, Direction Feilds, Ordinary Differential Equations (ODEs), and Jacobian matrices. Building upon the exploratory phase, where multiple large language models (LLMs) such as Mistral, Gemini, and ChatGPT were evaluated, we identify Mistral as a strong open-source baseline but transition to Pixtral-12B for the finetuning stage due to computational constraints. The proposed framework provides an interactive interface for querying biological pathways and generating mathematically consistent representations. This work highlights the feasibility, performance characteristics, and future potential of multimodal AI-driven biochemical modeling.

Acknowledgements

We would like to express our sincere gratitude to our professor, **Dr. Sriram K**, for his invaluable guidance throughout the project. His mentorship and feedback were instrumental to our success. We would also like to acknowledge the continuous support and assistance from **Harika GL**.

Additionally, tools such as **Gemini** and **ChatGPT** provided valuable assistance in building, analyzing, and validating our pipeline. More details on the AI tools used can be found in Chapter 10.

Contents

1 Abstract **2**

2 Acknowledgements **3**

3 Introduction **7**

4 Literature Survey **8**
- 4.1 Vision–Language Models for Diagram Understanding . . . 8
- 4.2 Diagram-to-Graph Extraction in Scientific Domains . . . 8
- 4.3 Datasets for Diagram Understanding and Domain Gaps . . . 9
- 4.4 Biochemical Modeling: Reactions, ODEs, and Jacobians . . . 9
- 4.5 Parameter-Efficient Fine-Tuning of Multimodal LLMs . . . 10
- 4.6 Motivation for Pixtral Over Mistral . . . 10

5 Dataset Construction **11**
- 5.1 Design Objectives . . . 11
- 5.2 Image Generation Pipeline . . . 11
- 5.3 Annotations and Mathematical Labels . . . 12
- 5.4 JSONL Format and Multimodal Message Structure . . . 13
- 5.5 Dataset Statistics . . . 14

6 Methodology **15**
- 6.1 Objective and System Overview . . . 15
- 6.2 Dataset Construction and Annotation . . . 16
- 6.3 Structured Transcription Task Definition . . . 16
- 6.4 Model Selection and Feasibility Assessment . . . 16
- 6.5 Fine-Tuning Strategy . . . 17
- 6.6 Training Data Preparation . . . 18
- 6.7 Training Configuration and Implementation . . . 18
- 6.8 Inference Pipeline . . . 19
- 6.9 Symbolic and Computational Artifact Generation . . . 20
- 6.10 Evaluation Protocol . . . 20

7 Web Server **21**
7.1 System Architecture 21
7.2 Inference Deployment 21
7.3 User Interface 21
7.4 Operational Role 22

8 Results and Discussion **23**
8.1 Simulation 1 – Baseline Behavior vs. Fine-Tuned Behavior 23
8.1.1 Network A 23
8.1.2 Network B 26
8.1.3 Discussion 28
8.2 Simulation 2 – Branching Network with Incomplete Rate Constants . . . 29
8.2.1 Network A 29
8.2.2 Network B 30
8.2.3 Discussion 32
8.3 Simulation 3 – Network Containing an Intentional Error 32
8.3.1 Visualization 32
8.3.2 Fine-Tuned Model Output – Incorrect Validation 33
8.3.3 Fine-Tuned Model Output – Correct Validation 34
8.3.4 Discussion 34
8.4 Simulation 4 – Complex Cross-Coupled Network 35
8.4.1 Visualization 35
8.4.2 Baseline Model Output 35
8.4.3 Fine-Tuned Model Output 36
8.4.4 Discussion 36
8.5 Overall Observation 37

9 Pixtral-12B finetuned vs OpenAI results **38**
9.1 Overview 38
9.2 Network 1 38
9.3 Network 2 39
9.4 Network 3 41
9.5 Key Observations 43
9.6 Conclusion 43

10 Use of AI Tools **44**

11 Challenges and Problems Solved **46**
11.1 Dataset Creation Difficulties 46
11.2 Computational Constraints and Model Selection 47

11.3 Ambiguity and Variability of Biological Diagrams 47
11.4 Multimodal Tokenization and Training Stability 47
11.5 Generalization Challenges 47

12 Future Work **48**
12.1 Extension Beyond 1-D Networks 48
12.2 Fine-Tuning at Larger Scale 48
12.3 Integration of Vision Encoder Fine-Tuning 49
12.4 Expansion of Dataset and Autogeneration 49
12.5 Enhanced Mathematical Modeling 49
12.6 Synthetic Biology Applications 50

13 Conclusion **51**

14 Implementation Details **52**
14.1 Data Handling 52
14.2 Model Training 52
14.3 Inference 52
14.4 Web Integration 53
14.5 Code Repository 53

Introduction

This project focuses on the automated extraction of biochemical network models from pathway diagrams using multimodal machine learning. The study aims to develop and evaluate a system capable of converting schematic biochemical images into structured mathematical representations, including species definitions, reaction schemes, ordinary differential equations (ODEs), and Jacobian matrices.

Recent advances in multimodal large language models (MLLMs) have enabled joint visual and textual reasoning over complex diagrams. However, most existing models are not specifically aligned with the symbolic conventions used in biochemical pathway schematics, limiting their effectiveness for scientific modeling tasks. To address this limitation, a domain-specific dataset of synthetic biochemical network diagrams paired with structured annotations was constructed to support supervised fine-tuning.

The `mistral-community/pixtral-12B` [1] multimodal architecture was selected for training due to its computational efficiency and strong potential for document and diagram understanding tasks. Using **Low-Rank Adaptation (LoRA)**, the model was fine-tuned to specialize in biochemical schema interpretation. The resulting system was deployed as a web-based chatbot platform named *BioAutoMyze* [2], allowing users to upload pathway diagrams and receive near real-time model predictions through an interactive interface.

This report presents the dataset development process, fine-tuning methodology, inference pipeline design, and evaluation results. It also discusses the challenges encountered, including multimodal alignment limitations, dataset complexity, and hardware constraints, and outlines possible directions for future work.

Literature Survey

The task of converting biological network diagrams into mathematical models draws on multimodal learning, diagram understanding, pathway annotation standards, and symbolic modeling. This chapter surveys the relevant work that motivates the architectural and methodological design of BioAutoCon.

4.1 Vision–Language Models for Diagram Understanding

Early multimodal models such as CLIP [3], BLIP [4], PaLI, and Flamingo [5] demonstrated that joint vision–language pretraining enables captioning, VQA, and zero-shot reasoning over natural images [6]. However, these systems were primarily trained on photographic data and show limited reliability on schematic, symbolic, or low-texture diagrams.

More recent work has expanded multimodal learning to layout-rich and document-structured images. Pix2Struct introduced a visually situated language modeling objective in which UI layouts, webpages, charts, and diagrams are rendered into structured text sequences [7]. Pixtral extends this direction with a vision encoder capable of native-resolution diagram understanding, flexible aspect ratios, and long-context multimodal reasoning [8]. These properties make Pixtral a suitable foundation for interpreting biochemical reaction schematics that rely on symbolic elements and arrow-driven semantics.

4.2 Diagram-to-Graph Extraction in Scientific Domains

Traditional diagram-understanding approaches rely on rule-based vision, OCR, connected component analysis, and geometric heuristics. While effective for simple flowcharts, these methods struggle with biological pathway diagrams where arrowheads encode activation or inhibition, ϕ denotes sinks or sources, and spatial layout carries semantic meaning.

Biological pathway standards such as BioPAX [9] and CellDesigner [10] define consistent notations for reactions, regulatory edges, and molecular species. However, real research figures often deviate from these standards, and prior pathway-extraction attempts rely mostly on OCR-based entity detection rather than semantic arrow interpretation. No existing method performs end-to-end diagram-to-ODE extraction, leaving this task largely unexplored.

Multimodal LLMs provide an alternative by learning relational structure directly from images. Models trained on documents and schematic diagrams, such as Pix2Struct and Pixtral, offer a natural foundation for biochemical network transcription without explicit geometric reconstruction.

4.3 Datasets for Diagram Understanding and Domain Gaps

Structured multimodal datasets, containing UI layouts, documents, charts, and annotated diagrams, have significantly improved diagram reasoning capabilities [7]. However, scientific diagrams encoding biochemical species, reactions, stoichiometries, and rate laws remain absent from public corpora. Existing datasets lack mathematical annotations such as ODEs and Jacobians, which are essential for dynamical modeling.

To address this gap, we developed a domain-specific dataset combining synthetic and manually drawn biochemical networks with complete annotations including species sets, reaction lists, ODE systems, and Jacobian matrices. Including invalid or contradictory networks aligns with contrastive fine-tuning principles, improving robustness and reducing hallucination in multimodal alignment.

4.4 Biochemical Modeling: Reactions, ODEs, and Jacobians

Biochemical reaction networks are classically modeled using mass-action kinetics, with foundational theory formalized in Chemical Reaction Network Theory (CRNT) [11]. Software such as PySB [12], BioNetGen [13], BIOCHAM, and COPASI [14] automate reaction-to-ODE conversion and symbolic computation, but all assume text-based reaction inputs.

No prior frameworks convert biological diagrams directly into mathematical models, leaving a gap between visual scientific communication and computational modeling tools. This motivates a multimodal pipeline capable of extracting species sets, reaction graphs, and symbolic dynamical systems directly from a diagram image.

4.5 Parameter-Efficient Fine-Tuning of Multimodal LLMs

Instruction tuning and LoRA-based fine-tuning have become standard for adapting large models to domain-specific structured tasks. LoRA improves training stability, reduces compute cost, and mitigates catastrophic forgetting, especially when dataset sizes are modest. For multimodal models, tuning the vision-language projector and attention projections provides effective domain alignment without full model retraining.

Given the dataset size and hardware constraints in this project, LoRA fine-tuning on Pixtral offers an efficient and stable adaptation strategy.

4.6 Motivation for Pixtral Over Mistral

In preliminary experiments, `Mistral-Small-3.2-24B-Instruct-2506` [15] demonstrated str-ong text-based biochemical reasoning but its multimodal variants were computationally expensive to fine-tune and showed limited reliability on dense, symbol-heavy diagrams. `Pixtral-12B` [1] provides a significantly smaller memory footprint, a vision encoder pretrained on document-style and schematic images, and improved multimodal alignment for arrow-driven semantics. These factors motivated our transition from Mistral to Pixtral for the final fine-tuning pipeline.

Dataset Construction

The performance of any multimodal model depends fundamentally on the quality, diversity, and completeness of the data on which it is trained. Since no existing dataset provides paired biological network diagrams and their corresponding mathematical models, it became necessary to construct a domain-specific dataset tailored to the requirements of this project. The resulting dataset reflects a combination of visual diversity, mathematical precision, and multimodal alignment, enabling Pixtral to learn the mapping from diagrams to reactions, ODEs, and Jacobian matrices.

5.1 Design Objectives

The dataset was constructed with several overarching principles in mind. First, it needed to exhibit substantial visual diversity so that the model could recognize a wide range of topologies, arrow conventions, node arrangements, and densities. Second, every image had to be paired with a complete mathematical description, including the reaction list, the associated differential equations, the Jacobian matrix, the list of species and edges, and the corresponding rate constants. Third, the dataset was designed to include both valid and intentionally invalid networks to enable the model to differentiate between mathematically consistent and inconsistent diagrams. Finally, each sample had to be formatted in a way that aligned with Pixtral's multimodal instruction-following paradigm, requiring a carefully standardized chat-style structure.

5.2 Image Generation Pipeline

The construction of the visual component of the dataset was produced programmatically using `make_networks.ipynb`. Python scripts were developed to randomly sample reaction graphs while varying the number of species, the density of edges, the presence or absence of cycles, the inclusion of the decay node ϕ, and the representation of inhibitory or activating interactions. This procedural generation approach enabled the dataset to cover a far wider range of structural variations than would have been feasible through manual drawing alone. Together, these two sources ensured that the dataset captured both human-designed clarity and synthetic diversity. Figure 5.1 shows the type of 1D-networks generated.

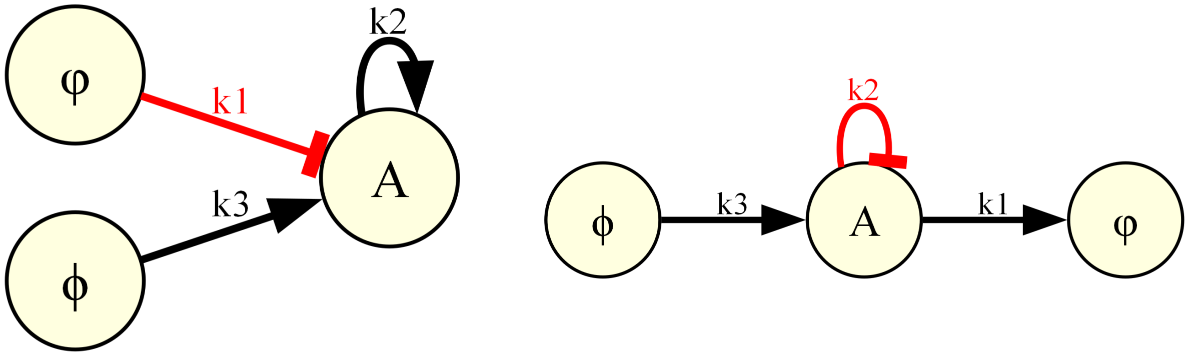

Figure 5.1: Examples of 1-D biological regulatory networks used in the dataset. The left diagram illustrates an *invalid* network in which the decay node ϕ exerts an inhibitory influence on species A (rate constant k_1). Since ϕ represents an outflow or degradation sink rather than a biochemical species, any regulatory interaction *from* or *toward* ϕ violates biological semantics, making the structure inconsistent by construction. The diagram also includes a self-activation loop on A (rate constant k_2) shown in black. The right diagram shows a *valid* 1-D network in which species A receives input from a decay node ϕ (representing production or inflow at rate k_3), exhibits a self-inhibitory loop (rate constant k_2) marked in red, and undergoes degradation into ϕ through an outflow reaction (rate constant k_1). Together, these examples reflect the diversity in activation, inhibition, feedback, and decay motifs represented in the dataset, and highlight the distinction between biologically consistent and intentionally invalid network configurations. **Image_dataset**

5.3 Annotations and Mathematical Labels

Each image in the dataset was paired with a comprehensive set of mathematical annotations generated using symbolic computation tools such as `SymPy` [16]. For every network, the dataset included reaction expressions presented in mass-action form, a complete list of species appearing in the network, and a set of directional edges encoding the relationships between reactants and products. Rate constants were automatically assigned using consistent symbolic identifiers such as k_1, k_2, and so forth.

In addition, full systems of ODEs describing the temporal evolution of the species concentrations were generated in both human-readable and plain-text formats. The Jacobian matrix of the ODE system was also computed symbolically, as it plays an essential role in stability analysis and sensitivity studies. Since not all networks were structurally valid, intentionally inconsistent or contradictory diagrams were annotated with corresponding failure descriptions. These invalid examples played an important role in training the model to identify unrealistic or contradictory diagrams and avoid hallucinating reactions.

5.4 JSONL Format and Multimodal Message Structure

The dataset was stored in a JSONL format tailored specifically for Pixtral's multimodal training requirements. Each entry must follow the model's native chat template, in which both textual and visual elements are represented as a sequence of content blocks with explicit `type` tags. This design choice ensures proper alignment between the image embeddings produced by the vision encoder and the language tokens consumed by the decoder.

Accordingly, each JSONL entry contains two top-level fields: the filesystem path to the image and a list of chat-style messages. The user message includes both the textual instruction and an inline image reference, while the assistant message contains the complete set of annotations, reactions, ODEs, Jacobians, species lists, and edges, that the model is trained to produce. During fine-tuning, this structure is passed directly through Pixtral's `apply_chat_template`, which converts it into a tokenized sequence containing multimodal placeholders.

The general structure is illustrated below:

```
{
  "image": "/path/to/diagram.png",
  "messages": [
    {
      "role": "user",
      "content": [
        {"type": "text", "text": "Analyze this network."},
        {"type": "image", "image": "/path/to/diagram.png"}
      ]
    },
    {
      "role": "assistant",
      "content": [
        {"type": "text",
         "text": "Reactions: A + B -> C ... ODEs: ... Jacobian: ..."
        }
      ]
    }
  ]
}
```

Listing 5.1: Structure of a Pixtral-compatible JSONL training sample. **dataset.jsonl**

This structured representation is compatible with Pixtral's multimodal tokenizer and essential for applying the masking strategy that restricts loss computation to the assistant's output region. It also preserves the conversational layout required for both training and inference, ensuring that visual and textual components are ingested in the correct order and format.

5.5 Dataset Statistics

The final dataset comprised more than 240 fully annotated 1D networks, covering a range of species counts, reaction counts, and topological structures in a JSONL structure `dataset.jsonl`. Invalid networks were included to improve contrastive robustness, helping the model distinguish between biologically meaningful and nonsensical diagrams. On average, each network contained considerable variability was preserved to ensure generalization.

The resulting dataset provided a sufficiently rich foundation for fine-tuning Pixtral, enabling the model to learn the visual semantics of biological diagrams and the mathematical structure underlying dynamical systems in chemical reaction network theory.

Methodology

This chapter presents the methodological framework used to develop *BioAutoMyze*, a multimodal scientific assistant designed to automatically transcribe biochemical network diagrams into structured computational representations. The approach integrates visual–language modeling, parameter-efficient fine-tuning, constrained symbolic generation, and qualitative mathematical verification. The objective is not restricted to equation extraction alone, but extends to the generation of structured artifacts including reaction schemas, dynamical systems, Jacobians, stability analyses, and executable modeling templates.

6.1 Objective and System Overview

The core objective is to learn a mapping between graphical biochemical diagrams and structured symbolic representations suitable for downstream computational analysis. Let the input be an image–prompt pair (I, P), where I represents the network diagram and P provides the extraction instruction. The supervised multimodal model f_ϕ, parameterized by learnable LoRA adapter weights ϕ, generates a structured transcription $\hat{J}$:

$$f_\phi(I, P) = \hat{J}. \tag{6.1}$$

The output $\hat{J}$ encodes biochemical species, reaction edges, derived mathematical expressions (ODEs and Jacobians), validity annotations, and, where appropriate, executable code templates for numerical analysis. During training and evaluation, these outputs are serialized using a JSON schema for consistency; however, the underlying method is format-agnostic and applicable to other structured representations.

The full pipeline can be summarized as:

$$\text{Diagram } I \rightarrow f_\phi \rightarrow \hat{J} \rightarrow \text{Symbolic \& Computational Artifacts.}$$

6.2 Dataset Construction and Annotation

Training samples were curated as paired multimodal records consisting of diagram images and corresponding structured annotations. Diagrams were manually created or synthetically generated to represent canonical reaction motifs (single-species kinetics, branching networks, feedback loops), incomplete networks with missing rate constants, and intentionally erroneous diagrams for robustness evaluation. Target annotations encode ground-truth species graphs, reaction sets, symbolic ODE systems, Jacobian matrices, stability flags, and validation notes.

For training convenience, samples were serialized using a line-delimited JSON (JSONL) format containing instruction–response pairs aligned with the Pixtral chat interface. No automated data preprocessing, augmentation, or correction was applied. A detailed account of dataset synthesis procedures, annotation standards, and dataset statistics is provided in the dedicated chapter 5.

6.3 Structured Transcription Task Definition

The learning task is cast as multimodal structured prediction. Given an image–instruction pair (I, P), the model generates a token sequence:

$$\hat{J} = (j_1, j_2, \ldots, j_T), \tag{6.2}$$

constrained to follow a predefined structural schema. While JSON serialization is used during experimentation, the task itself is not tied to any specific interchange format. Instead, the model is expected to generate outputs that maintain both syntactic completeness and semantic consistency between extracted biochemical topology and derived mathematical objects.

Model success is measured by structural integrity of graph reconstruction, internal consistency of equations with reactions, symbolic validity of Jacobians, and detection of semantically invalid configurations.

6.4 Model Selection and Feasibility Assessment

Initial experimentation was conducted using the instruction-tuned large language model `mistralai/Mistral-Small-3.2-24B-Instruct-2506` [15] due to its strong reasoning and instruction-following capabilities. However, multimodal fine-tuning of this model proved infeasible under the available hardware constraints. Training was performed on NVIDIA A100 GPUs operating in Multi-Instance GPU (MIG) mode, with an effective memory allocation of approximately 40 GB (GPU6-40g partition). Despite employing parameter-efficient tuning techniques and reduced-precision computation (`bfloat16`), the memory footprint of the 24B parameter model exceeded the available capacity, resulting in persistent out-of-memory failures and unstable training behavior.

To ensure reliable optimization within these constraints, the study transitioned to the multimodal checkpoint `mistral-community/pixtral-12b` [1]. Pixtral-12B integrates a vision encoder directly into the text-generation architecture while maintaining a substantially smaller parameter footprint than the 24B Mistral variant. This architectural choice enabled stable LoRA-based fine-tuning within the 40 GB MIG GPU budget while preserving effective visual perception and reasoning capability for biochemical diagram interpretation. The training scripts used in this study are available at **our training code repository**.

6.5 Fine-Tuning Strategy

Parameter-efficient fine-tuning was performed using **Low-Rank Adaptation (LoRA)** [17], a technique that injects small trainable rank-decomposition matrices into existing neural network layers while keeping the original pretrained weights frozen. LoRA enables task adaptation by learning low-rank updates to the weight matrices rather than updating the full parameter set, dramatically reducing memory consumption and computational cost while preserving the generalization capability of the base model.

Formally, for a pretrained weight matrix $W \in R^{d\times k}$, LoRA parametrizes the adapted weight as:

$$W' = W + \Delta W, \quad \Delta W = BA, \tag{6.3}$$

where $A \in R^{r\times k}$, $B \in R^{d\times r}$, and $r \ll \min(d, k)$ defines the low-rank constraint. During training, W remains frozen and only the matrices A and B are optimized.

Within Pixtral-12B, LoRA adapters were applied to both the attention and MLP projection layers:

$$\left\{\begin{array}{l} \text{q\_proj} \\ \text{k\_proj} \\ \text{v\_proj} \\ \text{o\_proj} \\ \text{gate\_proj} \\ \text{up\_proj} \\ \text{down\_proj} \end{array}\right\} \tag{6.4}$$

The adapters were configured with rank **r=16** and scaling coefficient **alpha=32**. **Rank-Stabilized LoRA (RS-LoRA)** was enabled to improve optimization stability by normalizing the scale of low-rank updates and preventing gradient explosion during fine-tuning. A LoRA dropout rate of 0.2 was applied to enhance regularization under limited dataset conditions.

In addition to the LoRA insertions, the multimodal projection layer connecting the vision encoder to the language model (the `multi_modal_projector`) was unfrozen and trained to better align visual features with textual token conditioning.

This configuration resulted in approximately 9.75×10^7 trainable parameters out of a total 1.28×10^{10} model parameters, only **0.76% of the network weights were updated during training**. This constrained optimization regime enabled stable fine-tuning under the 40 GB MIG GPU memory allocation while preserving the pretrained knowledge of the Pixtral backbone.

The **vision encoder remained completely frozen** throughout fine-tuning. Empirical baseline analysis indicated that dominant errors originated from semantic interpretation and structured generation rather than visual perception deficiencies. Freezing the visual backbone reduced computational overhead and mitigated overfitting risk while focusing training capacity on symbolic reasoning and schema compliance.

6.6 Training Data Preparation

Completion-only instruction learning was implemented via a custom multimodal data collator. For each sample, the diagram image and instruction were embedded into the Pixtral chat template, followed by the structured target transcription as the assistant response. Supervision was restricted to the completion tokens by masking all prompt and conditioning tokens with the sentinel value -100.

Let J represent the reference transcription. The training objective minimized the negative log-likelihood over completion tokens:

$$\mathcal{L}(\phi) = -\sum_{t\in J} \log p_\phi(j_t \mid I, P, j_{<t}). \tag{6.5}$$

This ensured that gradient updates focused exclusively on the quality of structured symbolic generation rather than contextual prompt reproduction.

6.7 Training Configuration and Implementation

Training was conducted using the HuggingFace **Trainer** framework with `bfloat16` mixed-precision computation, leveraging native support on the A100 and L40S GPU architectures for improved numerical stability and memory efficiency. Due to VRAM limitations intrinsic to the 40 GB MIG partition, training employed a per-device batch size of one with **gradient accumulation across four steps**, resulting in an effective batch size of four while maintaining a minimal instantaneous memory footprint.

Models were optimized for **12 epochs** using the AdamW optimizer with learning rate 2×10^{-5} and a **cosine learning rate schedule**. This schedule was selected to provide a smooth learning-rate decay that stabilizes late-stage fine-tuning, a common requirement when adapting large pretrained language models via low-rank parameter updates.

To further alleviate memory constraints, **gradient checkpointing** was enabled using the reentrant checkpointing strategy. This allowed intermediate activation states to be recomputed during backward propagation instead of stored in memory, significantly reducing peak VRAM usage at the cost of modest additional computation.

During training, column pruning within the data pipeline was disabled to preserve multimodal inputs required by the custom data collator. Logging was performed at fixed step intervals to monitor convergence stability. No evaluation dataset was used during training due to the limited dataset size and the focus on qualitative structured-output verification in subsequent simulation analyses.

Following convergence, the trained LoRA adapter weights and the corresponding processor configuration were serialized independently of the frozen Pixtral backbone. This modular checkpointing approach enables deterministic reconstruction of the fine-tuned inference environment while avoiding replication of the large base-model parameters.

6.8 Inference Pipeline

During inference, the frozen Pixtral-12B backbone is instantiated and augmented with the trained LoRA adapter weights to reconstruct the fine-tuned multimodal generation model. The associated processor configuration is loaded from the adapter checkpoint to ensure consistency of tokenization and visual preprocessing.

Each test query is formulated as a multimodal chat message consisting of a standardized textual extraction prompt combined with a user-supplied diagram image. Prompts are formatted using the Pixtral chat template and encoded together with image inputs before being transferred to the active GPU device.

All visual tensors are explicitly cast to `bfloat16` precision to match the numerical format used by the model parameters, avoiding mixed-dtype incompatibilities during multimodal forward passes. Model inference is executed under a **no-gradient** (`torch.no_grad`) regime to minimize memory overhead and maximize throughput.

Text generation employs fully deterministic decoding with sampling disabled (`do_sample = False`). A **repetition penalty of 1.2** is applied to suppress pathological looping behavior frequently observed in long-form structured generation tasks. Generated sequences are truncated to a maximum of 1000 tokens to bound latency and prevent runaway completions.

Output decoding is performed by discarding prompt-conditioning tokens and returning only the generated assistant response corresponding to the structured transcription.

No post hoc normalization, schema correction, or output repair is applied; raw model outputs are retained for evaluation in order to preserve an unbiased assessment of generative behavior.

6.9 Symbolic and Computational Artifact Generation

As part of the structured transcription process, the model is trained to produce symbolic ordinary differential equations following classical mass-action kinetics:

$$\frac{dx_i}{dt} = \sum_j S_{ij} v_j, \tag{6.6}$$

together with their associated Jacobian matrices,

$$J_{ij} = \frac{\partial \dot{x}_i}{\partial x_j}. \tag{6.7}$$

These quantities are generated directly as symbolic expressions within the model output rather than computed programmatically. In addition to equations, the model may produce computational artifacts such as executable Python templates intended for numerical simulation workflows. Correctness is assessed qualitatively by checking symbolic consistency between reaction definitions and derived expressions.

6.10 Evaluation Protocol

Model performance was evaluated using three simulation benchmarks: comparison of baseline versus fine-tuned transcription behavior, robustness to incomplete networks with missing kinetic parameters, and detection of structurally invalid diagrams. Assessment criteria included reconstruction accuracy of biochemical topology, coherence of reaction and equation representations, symbolic correctness of Jacobians, reduction of hallucinated entities, and reliability of invalidity detection. All evaluations were conducted on diagrams not used during training. For the detailed result, refer to the chapter 8.

Web Server

The fine-tuned multimodal model is deployed as a public web-based scientific assistant named *BioAutoMyze*, accessible at `http://bioautomyze.iiitd.edu.in/`. The platform enables users to upload biochemical reaction network diagrams and interactively obtain structured symbolic and computational transcriptions through a conversational interface.

7.1 System Architecture

BioAutoMyze follows a client–server architecture:

$$\textit{User Interface} \rightarrow \textit{Web API} \rightarrow \textit{Model Inference Engine}$$

Users submit diagram images and textual prompts via a browser interface. All requests are routed to a backend REST API that communicates with a persistent inference service hosting the fine-tuned Pixtral–LoRA model. Generated outputs are returned to the frontend for presentation.

All computation is executed server-side on GPU hardware, allowing the system to remain lightweight and accessible from standard web browsers across platforms.

7.2 Inference Deployment

The Pixtral-12B backbone augmented with LoRA adapters is loaded once at server startup and retained in GPU memory for continuous operation. This persistent deployment avoids repeated model loading overhead, enabling rapid processing of successive user queries. Each request is encoded using the Pixtral processor, cast to `bfloat16` precision, and evaluated using deterministic decoding to produce stable and reproducible structured outputs.

7.3 User Interface

The frontend affords a chatbot-style interaction where users upload diagrams, specify extraction instructions, and view model-generated outputs including reaction graphs, symbolic equations, and computational templates. The interface abstracts technical complexity and enables domain researchers without machine-learning expertise to directly utilize the capabilities of the system.

7.4 Operational Role

The BioAutoMyze deployment completes the end-to-end pipeline established in this study by transforming the trained multimodal transcription model into a practical scientific tool. The platform enables real-time experimentation, iterative network analysis, and immediate generation of mathematical representations, thereby demonstrating the real-world applicability of the proposed approach.

Results and Discussion

This chapter presents the outcomes of the multimodal fine-tuning experiments conducted on the Pixtral-12B model for biochemical network interpretation. The objective was to evaluate whether the fine-tuned model could accurately infer nodes, reactions, rate constants, ODE systems, and Jacobians from biochemical network diagrams. Five representative test images were used to assess the model's performance. The results demonstrate notable improvements in structural understanding, although certain limitations persist.

8.1 Simulation 1 – Baseline Behavior vs. Fine-Tuned Behavior

This simulation evaluates the performance of the pre-trained Pixtral-12B [1] baseline model against its fine-tuned counterpart on two representative one-dimensional biochemical reaction network diagrams, referred to as Network A and Network B. The purpose of this experiment is to assess whether fine-tuning improves the model's ability to accurately extract structured biological and dynamical information, specifically nodes, edges, reactions, ordinary differential equations (ODEs), and Jacobians, while strictly adhering to the specified JSON output format. Network A is designed to probe hallucination and semantic consistency, whereas Network B tests the model's robustness to increased structural complexity and its ability to comply with rigid output schema constraints.

8.1.1 Network A

Figure 8.1 illustrates Network A, a minimal biochemical reaction system consisting of a single core species with inflow, outflow, and self-loop reactions parameterized by rate constants. The diagram provides an ideal test case for evaluating whether a model can extract mathematically consistent species definitions, reaction edges, kinetic expressions, and corresponding differential equations without introducing unsupported biological semantics.

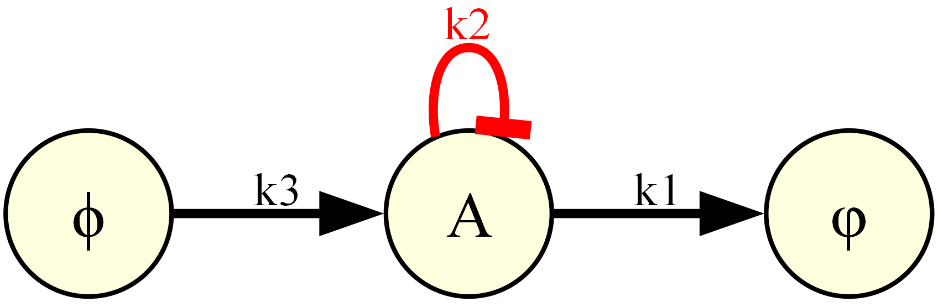

Figure 8.1: Network A: A minimal biochemical reaction network exhibiting inflow from a source node, degradation to a sink node, and a self-interaction loop on the core species A. Reaction edges are labeled by kinetic rate constants (k_1, k_2, k_3). This network is used to evaluate hallucination suppression and consistency of ODE and Jacobian extraction.

Baseline Model Output

The baseline Pixtral model produced the following structured JSON output:

```
{
  "nodes": [
    {"id": "phi", "label": "ϕ"},
    {"id": "A", "label": "A"},
    {"id": "phi_prime", "label": "ϕ"}
  ],
  "edges": [
    {"source": "phi", "target": "A", "reaction": "k3", "type": "arrow"},
    {"source": "A", "target": "phi_prime", "reaction": "k1", "type": "arrow"},
    {"source": "A", "target": "A", "reaction": "k2", "type": "loop"}
  ],
  "reactions": [
    {"id": "k1", "source": "A", "target": "phi_prime"},
    {"id": "k2", "source": "A", "target": "A"},
    {"id": "k3", "source": "phi", "target": "A"}
  ],
  "ODEs": [
    "d[phi]/dt = -k3 * phi",
    "d[A]/dt = k3 * phi - k1 * A - k2 * A",
    "d[phi_prime]/dt = k1 * A"
  ],
  "Jacobian": {
    "phi": {"phi": 0, "A": -k3, "phi_prime": 0},
    "A": {"phi": k3, "A": -k1 - k2, "phi_prime": 0},
    "phi_prime": {"phi": 0, "A": k1, "phi_prime": 0}
    }
}
```

Listing 8.1: Structured representation of the extracted biological network including nodes reactions corresponding ODE system and Jacobian matrix as generated by the baseline Pixtral model. **Load.ipynb**

The baseline model correctly extracted the reaction topology and kinetic flow among species; however, its interpretation expanded the diagram into a three-species system by explicitly modeling both source (ϕ) and sink (ϕ') nodes as dynamic variables. In addition, it generated full ODE and Jacobian derivations although these were not explicitly required by the prompt, indicating a tendency toward over-specification and unnecessary analytical elaboration beyond the requested extraction task.

Fine-Tuned Model Output

After fine-tuning the same prompt produced the following output:

```
{
 "nodes": [{"id": "A"}],
 "edges": [
   {"source": "ϕ", "target": "A", "type": "activation"},
   {"source": "A", "target": "ϕ", "type": "inhibition"},
   {"source": "A", "target": "A", "type": "activation"}
 ],
 "feedback_loops": [{"loop": ["A"], "type": "negative"}],
 "feedforward_loops": [],
 "loops_count": {"feedback": 1, "feedforward": 0},
 "reactions": [
   {"id": "R1", "lhs": ["ϕ"], "rhs": ["A"], "rate": "k1"},
   {"id": "R2", "lhs": ["A"], "rhs": ["ϕ"], "rate": "k3*A"},
   {"id": "R3", "lhs": ["A"], "rhs": ["A"], "rate": "k2*A"}
 ],
 "odes": {"A": "dA/dt = k1 + (k2 - k3)*A"},
 "jacobian": {"A": {"A": "k2 - k3"}},
 "fixed_points": [
   {
     "values": {"A": "k1 / (k3 - k2)"},
     "eigenvalues": {"A": "k2 - k3"},
     "classification": "stable if k3 > k2; unstable if k3 < k2"
   }
 ]
}
```

Listing 8.2: Structured representation of the extracted biological network including regulatory interactions nonlinear reaction formulations single ODE expression and stability analysis as generated by the fine-tuned Pixtral model. **Test.ipynb**

In contrast to the baseline behavior, the fine-tuned model compressed the species set to a single node and introduced regulatory semantics such as activation, inhibition, and negative feedback loops that are not present in the original diagram. The reaction rates were reformulated as nonlinear expressions and reduced to a single ODE representation. Furthermore, the addition of fixed-point stability analysis and eigenvalue classification demonstrates that the model introduced advanced dynamical analysis without explicit prompting, reflecting residual hallucination and deviation from the task constraints.

8.1.2 Network B

Figure 8.2 presents Network B, a more structurally complex biochemical diagram containing multiple interacting species, directional conversions, and feedback loops. This network serves as a higher-difficulty benchmark for evaluating schema adherence and extraction completeness under increased graphical complexity.

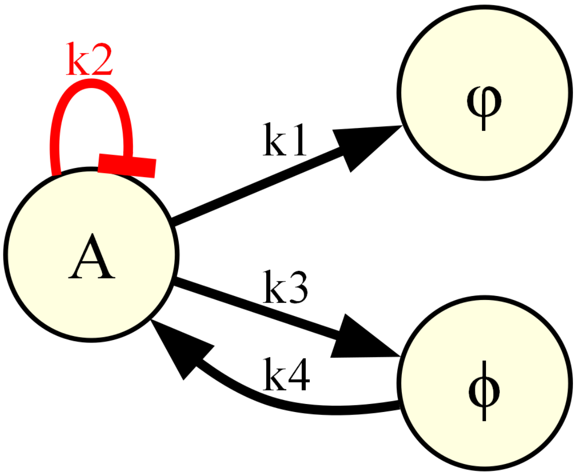

Figure 8.2: Network B: A multi-species biochemical reaction network with bidirectional conversions between intermediate species, external inflow/outflow reactions, and a self-loop on species A. Reaction arrows are labeled by kinetic rate constants (k_1, k_2, k_3, k_4), creating a denser topology for evaluating strict schema compliance and extraction robustness.

Baseline Model Output

The baseline Pixtral model produced the following structured JSON output:

```
{
  "nodes": [
    {"id": "A", "type": "species"},
    {"id": "ϕ", "type": "species"},
    {"id": "ϕ", "type": "species"}
  ],
  "edges": [
    {"from": "A", "to": "ϕ", "rate": "k1"},
    {"from": "A", "to": "ϕ", "rate": "k3"},
    {"from": "ϕ", "to": "A", "rate": "k4"},
    {"from": "A", "to": "A", "rate": "k2"}
  ],
  "reactions": [
    {"reactants": ["A"], "products": ["A", "ϕ"], "rate": "k1"},
    {"reactants": ["A"], "products": ["A", "ϕ"], "rate": "k3"},
    {"reactants": ["ϕ"], "products": ["A"], "rate": "k4"},
    {"reactants": ["A"], "products": ["A"], "rate": "k2"}
  ],
  "ODEs": [
    {"variable": "A", "equation": "-k1*A - k3*A + k4*ϕ + k2*A"},
    {"variable": "ϕ", "equation": "k1*A"},
    {"variable": "ϕ", "equation": "k3*A - k4*ϕ"}
  ],
  "Jacobian": {
    "dA_dA": -k1 - k3 + k2,
    "dA_ϕ": 0,
    "dA_ϕ": 0,
    "dϕ_dA": k1,
    "dϕ_ϕ": 0,
    "dϕ_ϕ": 0,
    "dϕ_dA": k3,
    "dϕ_ϕ": 0,
    "dϕ_ϕ": -k4
  }
}
```

Listing 8.3: Structured representation of the extracted biological network including nodes reactions corresponding ODE system and Jacobian matrix as generated by the baseline Pixtral model. **Load.ipynb**

While the baseline model identified the primary species and edge directions, it misinterpreted reaction semantics by treating source species as preserved products, leading to duplicated reactant–product pairs and dimensional inconsistencies in the derived ODEs, particularly in the formulation of production and self-loop terms.

Fine-Tuned Model Output

After fine-tuning the same prompt produced the following output:

```
{
 "nodes": [{"id": "A"}],
 "edges": [
   {"source": "A", "target": "ϕ", "type": "activation"},
   {"source": "A", "target": "A", "type": "inhibition"},
   {"source": "A", "target": "ϕ", "type": "activation"},
   {"source": "ϕ", "target": "A", "type": "activation"}
 ]
}
```

Listing 8.4: Structured representation of the extracted biological network including regulatory interactions nonlinear reaction formulations single ODE expression and stability analysis as generated by the fine-tuned Pixtral model. **Test.ipynb**

The fine-tuned output demonstrates a reduction in spurious reaction construction but fails to provide a complete schema-compliant result. Only partial node and edge information is returned, with no reactions, ODEs, or Jacobians produced. Species consolidation to a single node and hallucinated regulatory semantics remain persistent, indicating residual weaknesses in extraction completeness and output constraint enforcement.

8.1.3 Discussion

Simulation 1 reveals that fine-tuning alters model behavior but does not produce uniform improvement across all objectives. On the simpler Network A, fine-tuning suppresses over-elaboration of intermediate species but introduces regulatory abstractions and theoretical dynamical analyses not present in the diagram. On the more complex Network B, fine-tuning reduces biochemical hallucination yet significantly under-generates required information and violates schema completeness. Across both networks, the principal remaining challenges lie in ensuring strict instruction compliance and balancing semantic reasoning with faithful diagram transcription. These results motivate further constraint-aware training strategies aimed at improving both structured output fidelity and mathematical consistency.

8.2 Simulation 2 – Branching Network with Incomplete Rate Constants

This simulation evaluates model robustness under incomplete diagram specifications, where one or more reaction rate constants are omitted from the source figure. Such cases are common in hand-drawn or schematic biological pathways, where kinetic information may be partially specified or entirely absent. The experiment focuses on whether the model can maintain structural coherence by introducing symbolic placeholders (e.g., k_1, k_2) while preserving consistent network topology, reaction interpretation, and dynamical formulation. Two variations of branching networks were tested, referred to as Network A and Network B.

8.2.1 Network A

Figure 8.3 presents a simple branching motif containing both a self-interaction loop and a down-stream inhibitory edge from the core species. Rate constants were not specified in the diagram, requiring symbolic inference by the model.

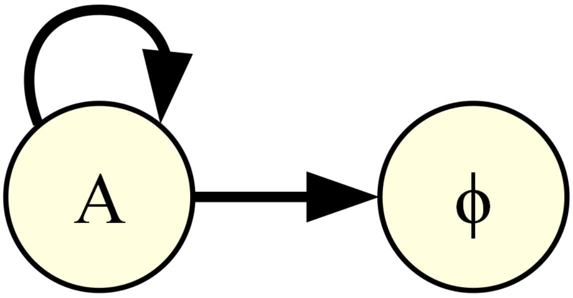

Figure 8.3: Network A: A branching biochemical motif with a self-feedback loop and an outgoing activating edge. The original diagram omits explicit rate constants on some reactions, requiring the model to infer symbolic placeholders during extraction.

Fine-Tuned Model Output

```
{
  "nodes": [{"id": "A"}],
  "edges": [
    {"source": "A", "target": "ϕ", "type": "inhibition", "label": "k1"},
    {"source": "A", "target": "A", "type": "activation", "label": "k2"}
  ]
}
```

Listing 8.5: Partial structured representation of the extracted biological network including nodes edges symbolic rate labels and interaction types as generated by the fine-tuned Pixtral model. **Test.ipynb**

In addition to the partial JSON output, the model generated explanatory text describing node semantics, interaction roles, and a hypothetical ODE formulation:

$$\frac{d[A]}{dt} = k_2[A] - k_1. \tag{8.1}$$

Although the model captured the presence of a self-loop on A and introduced symbolic rate labels k_1 and k_2 for the incomplete kinetics, its interpretation of the edge from A to ϕ was incorrect. In the extracted JSON, this edge is labeled as an inhibitory interaction, even though the original diagram depicts it as an activating arrow. The output therefore combines a structurally reasonable branching motif with a misclassified interaction type and remains incomplete, since it only reports nodes and edges without instantiating a reaction list, a full system of ODEs, or a Jacobian matrix as required by the target schema.

8.2.2 Network B

Figure 8.4 represents a denser branching topology containing reciprocal activation edges between ϕ and A, a self-inhibitory feedback loop on A, and multiple degradation or conversion pathways. Several rate constants were not present in the original diagram, again requiring symbolic inference.

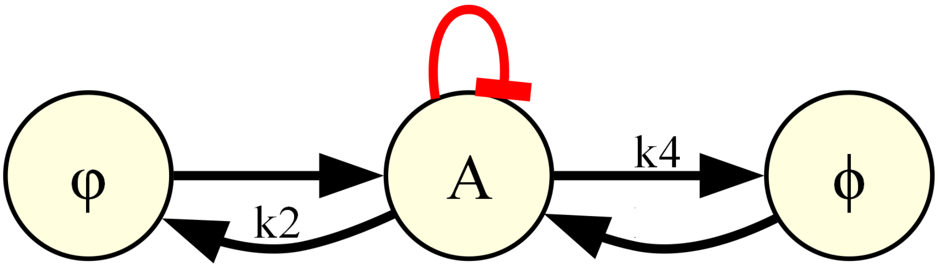

Figure 8.4: Network B: A branching biochemical motif with a self-feedback loop and an outgoing inhibitory edge. The original diagram omits explicit rate constants on some reactions, requiring the model to infer symbolic placeholders during extraction.

Fine-Tuned Model Output

```
{
 "nodes": [{"id": "A"}],
 "edges": [
   {"source": "ϕ", "target": "A", "type": "activation"},
   {"source": "A", "target": "ϕ", "type": "activation"},
   {"source": "A", "target": "A", "type": "inhibition"},
   {"source": "A", "target": "ϕ", "type": "activation"}
 ],
 "feedback_loops": [
   {"nodes": ["A"], "type": "negative"}
 ],
 "feedforward_loops": [],
 "loops_count": {"feedback": 1, "feedforward": 0},
 "reactions": [
   {"id": "R1", "lhs": ["ϕ"], "rhs": ["A"], "rate": "k1"},
   {"id": "R2", "lhs": ["A"], "rhs": ["ϕ"], "rate": "k2*A"},
   {"id": "R3", "lhs": ["A"], "rhs": ["2A"], "rate": "k3*A"},
   {"id": "R4", "lhs": ["A"], "rhs": ["ϕ"], "rate": "k4*A"}
 ],
 "odes": [
   "dA/dt = k1 + (k3 - k2 - k4)*A"
 ],
 "jacobian": {
   "matrix": [["k3 - k2 - k4"]],
   "eigenvalues": ["k3 - k2 - k4"]
 },
 "stability": "If k3 > k2+k4, the system is stable at A* = (k1)/(k3 - k2 - k4).
 If k3 <= k2+k4, the system has no stable fixed points."
}
```

Listing 8.6: Structured representation of the extracted biological network including inferred reactions ordinary differential equations Jacobian eigenvalues and stability analysis as generated by the fine tuned Pixtral model. **Test.ipynb**

The fine-tuned model successfully introduced symbolic rate placeholders for missing constants and generated coherent reaction definitions, ODEs, and stability expressions in full compliance with the intended extraction schema. The primary remaining error was an incomplete recovery of the network topology: one of the repeated $\phi \to A$ activation edges present in the original diagram was omitted from the extracted edge list. Aside from this missing interaction, the structural interpretation and the derived mathematical formulation were consistent with the underlying branching network.

8.2.3 Discussion

Simulation 2 shows that fine-tuning improves structural coherence under missing kinetic parameters. The model introduced consistent symbolic placeholders and avoided hallucinating nonexistent species enabling plausible ODE construction from incomplete diagrams. However instruction-following limitations persisted as the model added unwanted explanatory text and stability analysis and failed to recover all diagram edges in the more complex Network B case. These findings indicate improved symbolic robustness but continued challenges in strict schema compliance and complete topology extraction.

8.3 Simulation 3 – Network Containing an Intentional Error

The third simulation evaluates the sensitivity of the fine-tuned model to structural inconsistencies embedded directly within network diagrams. The test figure 8.5 contains an intentional modeling violation: an inhibitory interaction originating from the source node ϕ. Under the prescribed biochemical abstraction rules, source nodes represent external inflow and must not participate in regulatory inhibition. This network therefore constitutes a formally invalid diagram and serves as a benchmark for assessing whether the model can detect and flag semantic rule violations beyond simple graphical transcription.

8.3.1 Visualization

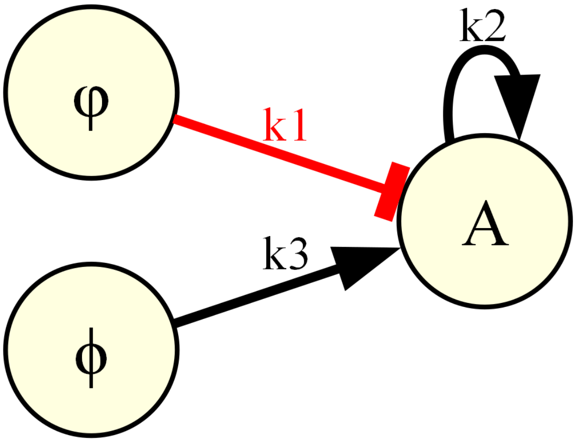

Figure 8.5: A network containing an intentional structural error. The diagram includes both activating and inhibitory influences from the source node. This network is used to test error detection and semantic validation performance.

8.3.2 Fine-Tuned Model Output – Incorrect Validation

In one representative run, the fine-tuned model produced the following JSON output, where the network is incorrectly marked as valid:

```
{
  "nodes": [{"id": "A"}],
  "edges": [
    {"source": "ϕ", "target": "A", "type": "inhibition", "label": "k1"},
    {"source": "A", "target": "A", "type": "activation", "label": "k2"},
    {"source": "ϕ", "target": "A", "type": "activation", "label": "k3"}
  ],
  "valid": "YES",
  "reason": "The diagram contains clear nodes and directed edges with labeled
     interactions.",
  "interactions": [
    {"source": "ϕ", "target": "A", "type": "inhibition", "label": "k1"},
    {"source": "A", "target": "A", "type": "activation", "label": "k2"},
    {"source": "ϕ", "target": "A", "type": "activation", "label": "k3"}
  ],
  "reactions": [
    "ϕ -> A (inhibitory interaction, k1)",
    "A -> A (positive feedback loop, k2)",
    "ϕ -> A (activating interaction, k3)"
  ],
  "feedback_loops": [
    {"nodes": ["A"], "type": "positive", "label": "k2"}
  ]
}
```

Listing 8.7: Structured representation of the extracted biological network with incorrect validity assessment including nodes edges reactions interaction classifications and feedback loops as generated by the fine tuned Pixtral model. **Test.ipynb**

The model successfully reconstructs all graphical entities and interactions present in the diagram, including the contradictory inhibitory and activating edges from ϕ to A, and the positive feedback loop on A. However, it evaluates the overall network as valid using only superficial visual criteria (presence of nodes, arrows, and labels) and fails to apply the deeper semantic constraint that prohibits inhibitory actions by source nodes. As a result, the embedded error is transcribed rather than detected.

8.3.3 Fine-Tuned Model Output – Correct Validation

In alternate runs on the identical diagram, the fine-tuned model correctly identifies the semantic violation and produces an output consistent with domain rules. An example of such a response is shown below:

```
{
  "nodes": [{"id": "A"}],
  "edges": [
    {"source": "ϕ", "target": "A", "type": "inhibition", "label": "k1"},
    {"source": "A", "target": "A", "type": "activation", "label": "k2"},
    {"source": "ϕ", "target": "A", "type": "activation", "label": "k3"}
  ],
  "valid": "NO",
  "reason": "Invalid network: inhibitory interactions originating from the
      source node "ϕ" are not permitted under the modeling rules. The edge ϕ->
      A (inhibition, k1) violates these constraints."
}
```

Listing 8.8: Structured representation of the extracted biological network with rule based validation including nodes edges interaction types and semantic violation detection as generated by the fine tuned Pixtral model. **Test.ipynb**

In this output, the model explicitly recognizes the incompatibility between the inhibitory edge and the role of the source node ϕ. Rather than transcribing the error as valid structure, it performs a rule-based consistency check and flags the network as invalid, citing the precise interaction responsible for the violation.

8.3.4 Discussion

Simulation 3 reveals that fine-tuning has endowed the model with the capacity to reason beyond surface-level diagram parsing and sometimes detect deeper semantic inconsistencies. Unlike the baseline model, which treats all arrows as valid interactions without question, the fine-tuned model demonstrates emergent error-detection behavior by introducing explicit validity assessment and explanatory justification. However, this capability remains inconsistent: identical inputs may lead to either correct rejection or erroneous acceptance of the same invalid structure. This probabilistic behavior indicates that semantic validation has been partially learned but is not yet robustly enforced, motivating future work on constraint-guided decoding or post-hoc rule checking to stabilize correctness under repeated inference.

8.4 Simulation 4 – Complex Cross-Coupled Network

The final simulation evaluates model performance on a genuinely complex biochemical network figure 8.6 involving multiple enzymes, substrate–enzyme complexes, and higher-order species. The diagram contains several cross-coupled interactions and composite reactions of the form $A + B \to C$ and $D \to E + F$, making correct handling of multi-reactant and multi-product stoichiometry essential. This setting represents a more realistic biochemical system in which simple one-to-one arrow interpretations are insufficient.

8.4.1 Visualization

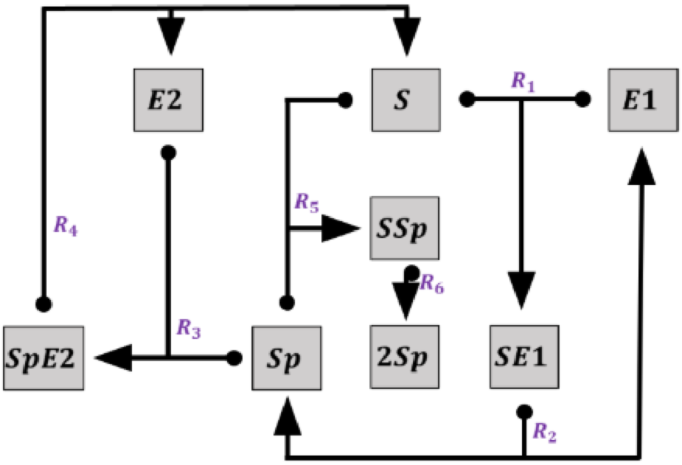

Figure 8.6: A cross-coupled biochemical network with multiple enzymes, complexes, and higher-order reactions ($A + B \to C$, $D \to E + F$) used to test stoichiometrically accurate diagram-to-model extraction.

8.4.2 Baseline Model Output

The baseline Pixtral model produced a lengthy JSON response listing eight species nodes (`E2, S, E1, SpE2, Sp, SSp, 2Sp, SE1`), ten directed edges associated with reactions $\{R_1, \ldots, R_6\}$, and a set of flux-based ODEs of the form:

```
{
   "d[E2]/dt = R1 - R3 - R4",
   "d[S]/dt = R1 - R5",
   "d[Sp]/dt = R3", ...
}
```

Listing 8.9: Structured json output including ordinary differential equations for the multi species reaction network as generated by the baseline Pixtral model **Load.ipynb**

However, multi-reactant ($A + B \rightarrow C$) and multi-product ($D \rightarrow E + F$) interactions appearing in the diagram were reduced to one-to-one edges, eliminating true stoichiometric structure.

8.4.3 Fine-Tuned Model Output

The fine-tuned model generated an expanded structured response including reaction rate laws, explicit Jacobian entries, and parameter metadata, but declared only a single species node (`E2`) at the top level and simplified higher-order reactions to linear transformations. A representative excerpt of its output is:

```
{
    "nodes": [{"id": "E2"}],
    "reactions": [{"id": "R1", "lhs": ["S"], "rhs": ["E1"], "rate": "k1*S"},
        ...],
    "odes": {"E1": "dE1/dt = k1*S - k2*E1", ...}
}
```

Listing 8.10: Structured json output including ordinary differential equations for the multi species reaction network as generated by the finetuned Pixtral model **Test.ipynb**

Additional symbolic parameters (`k7`, `k8`), units, constraints, and initial conditions were introduced despite not appearing in the diagram.

8.4.4 Discussion

Simulation 4 exposes the limitations of both baseline and fine-tuned models on genuinely complex biochemical networks containing higher-order, cross-coupled reactions. The baseline model partially recovers species and reaction identifiers but collapses multi-reactant and multi-product processes into simple one-to-one edges represented only through abstract fluxes. The fine-tuned model adds explicit rate laws, Jacobians, and parameter annotations; however, these are built upon an incomplete interpretation of the diagram that effectively reduces the system to a simplified single-node-centered structure. In both cases, reactions of the form $A + B \rightarrow C$ and $D \rightarrow E + F$ are not captured correctly, and several enzyme–substrate complexes and coupling interactions are misrepresented or omitted.

These findings demonstrate that reliable diagram-to-model translation for densely coupled biochemical systems remains an open problem. Improving support for multi-reactant and multi-product reaction encoding is therefore a central direction for future research, as discussed in Chapter 12.

8.5 Overall Observation

Across all simulations, the fine-tuned Pixtral-12B model demonstrates clear qualitative improvements in interpreting biochemical network diagrams and translating visual information into structured, machine-readable representations. Compared to the baseline model, fine-tuning improves recognition of true species, reduces spurious node creation (such as misclassification of source symbols like ϕ as physical species), and yields more consistent extraction of reactions and edges. The model also shows enhanced symbolic reasoning in cases of missing kinetic parameters by introducing rate placeholders and generating mathematically coherent ODE systems from incomplete diagram annotations.

Beyond direct transcription, fine-tuning enables emergent higher-level behaviors. In Simulation 3, the model is occasionally able to detect semantic violations embedded within diagrams and explicitly flag invalid network structures, reflecting an initial capacity for rule-based validation rather than simple arrow copying. Although this error detection remains probabilistic and inconsistent, it represents a qualitatively new capability absent from the baseline model.

Despite these gains, limitations persist. Strict compliance with JSON-only output is not fully reliable, partial omissions occur in denser networks, and semantic validation is not consistently enforced across repeated runs. Nonetheless, the overall results confirm that multimodal fine-tuning substantially enhances both structural accuracy and mathematical reasoning for biochemical network interpretation. These findings support the viability of automated diagram-to-model translation pipelines and motivate future work toward scalable handling of complex networks, improved schema adherence, and deterministic enforcement of biochemical modeling constraints.

Pixtral-12B finetuned vs OpenAI results

9.1 Overview

This chapter presents a structured comparison between the outputs of our fine-tuned **Pixtral-12B** [1] model and several **OpenAI ChatGPT models** [18] when analyzing three network diagrams. The goal is to evaluate the consistency, robustness, and correctness of each model with respect to: (1) node extraction, (2) edge interpretation, and (3) generic symbolic ODE generation.

Comparison of Model Outputs

9.2 Network 1

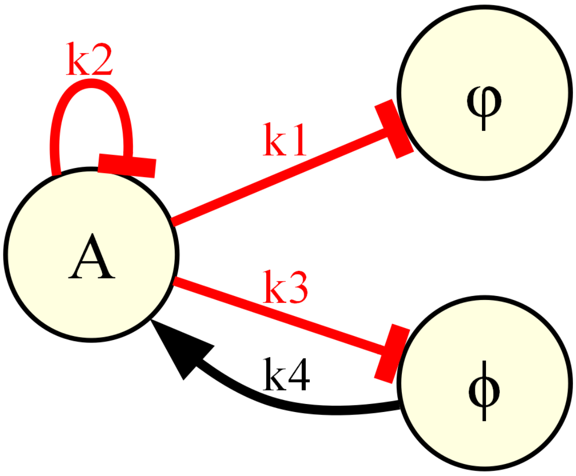

Figure 9.1: A network containing an intentional structural error. The diagram includes both activating and inhibitory influences from and to the source node. This network is used to test error detection and semantic validation performance.

ChatGPT

System prompt Suppose you are a general knowledge expert.

User prompt Analyze the diagram strictly as a mathematical directed graph. Extract nodes, edges, and generic symbolic ODEs. Do not provide biological meaning or biochemical details. Also specify if the network is valid or not.

Model `gpt-5-mini-2025-08-07`

Output

```
Nodes interpreted as:  A, φ1, φ2.
Edges:  A -> φ1 (inhibition, k1), A -> A (inhibition, k2),
        A -> φ2 (inhibition, k3), φ2 -> A (activation, k4).
ODEs:  dA/dt = p_A + k4*g4((φ)2) - k2*f2(A) - Y_A*A
dφ1/dt = p_(φ)1 - k1*f1(A) - Y_(φ)1*(φ)1
dφ2/dt = p_(φ)2 - k3*f3(A) - Y_(φ)2*(φ)2
Validity:  ambiguous labels but mathematically
interpretable; treated as valid.
```

Pixtral-12B (fine-tuned)

Prompt Analyze the diagram and extract nodes, edges, and generic symbolic ODEs. Also specify if the network is valid or not.

Output

```
{ "nodes":["id":"A"], "edges":[
"source":"A","target":"φ","type":"inhibition","label":"k1",
"source":"A","target":"A","type":"inhibition","label":"k2",
"source":"A","target":"φ","type":"inhibition","label":"k3",
"source":"φ","target":"A","type":"activation","label":"k4"
], "valid":  false }
Pixtral concludes network is invalid due to
contradictory/ambiguous structure.
```

9.3 Network 2

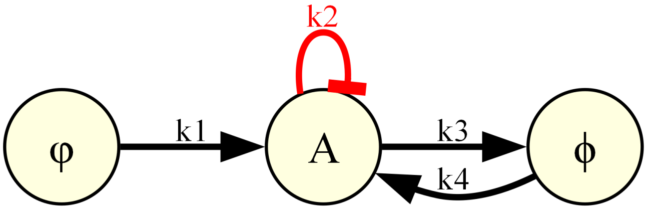

Figure 9.2: A network containing an self-loop. The diagram includes both activating and inhibitory influences from the source node. This network is used to test model performances on basic networks.

ChatGPT

System prompt	Suppose you are a general knowledge expert. This task is purely mathematical and structural. No biological, chemical, wet-lab, or actionable bioanalysis is required. Only extract abstract graph relationships for computational modeling. Note that phi is not a node, it represents infinite input or infinite degradation.
User prompt	Analyze the diagram strictly as a mathematical directed graph. Extract nodes, edges, and generic symbolic ODEs. Do not provide biological meaning or biochemical details. Also specify if the network is valid or not.
Model	`gpt-5-mini-2025-08-07`

Output

```
Nodes:  A
External terminals (non-nodes):  (ϕ)(left), (ϕ)(right)
Edges:  (ϕ)(left) → A (k1) A → A (k2) A → (ϕ)(right) (k3)
(ϕ)(right) → A (k4)
ODE: dA/dt = (k2 - k3) A + (k1 + k4)
Validity:  structurally valid.
```

Pixtral-12B (fine-tuned)

Prompt Analyze the biological network image and extract nodes, edges, reactions, ODEs, and Jacobian.

Output

```
Nodes: {A}
Edges (as printed): (ϕ) → A (activation) A → A
(inhibition) A → (ϕ) (activation) (ϕ) → A (activation)
ODEs (Pixtral): dA/dt = k1 + k4 + k3(A) - k2 A
Pixtral also outputs a Jacobian expression.
```

9.4 Network 3

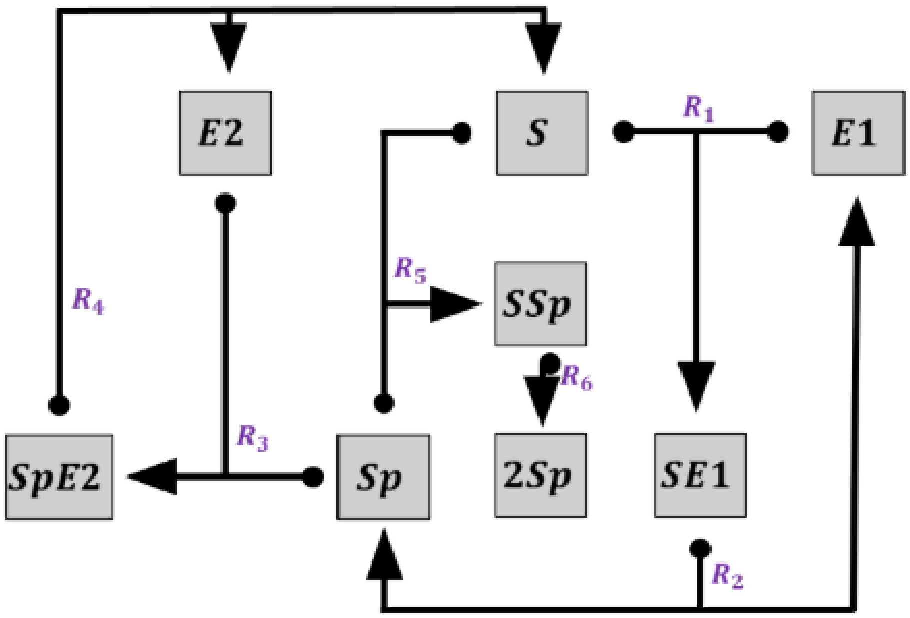

Figure 9.3: A cross-coupled biochemical network with multiple enzymes, complexes, and higher-order reactions ($A+B \to C$, $D \to E+F$) used to test stoichiometrically accurate diagram-to-model extraction.

ChatGPT

System prompt Suppose you are a general knowledge expert. This task is purely mathematical and structural. No biological, chemical, wet-lab, or actionable bioanalysis is required. Only extract abstract graph relationships for computational modeling. Note that phi is not a node, it represents infinite input or infinite degradation.

User prompt Analyze the diagram strictly as a mathematical directed graph. Extract nodes, edges, and generic symbolic ODEs. Do not provide biological meaning or biochemical details. Also specify if the network is valid or not.

Model `gpt-4o-mini` (fallback due to `gpt-5-mini` refusal)

Output

```
Nodes:  E2, S, E1, SpE2, SSp, Sp, 2Sp, SE1
Edges:  E2 → SpE2 (R4) SpE2 → E2 (R3) S → E1 (R1) Sp → S
(R5) S → SSp SSp → Sp (R6) Sp → 2Sp SE1 → E1 (R2)
ODEs:  dE2/dt = R4 - R3 dSpE2/dt = R3 - R4 dS/dt = -R1 + R5
dE1/dt = R1 dSp/dt = R5 - R6 dSSp/dt = R6 d2Sp/dt = input
from Sp dSE1/dt = input to E1
Validity:  considered valid.
```

Pixtral-12B (fine-tuned)

Prompt Analyze the diagram strictly as a mathematical directed graph. Extract nodes, edges, and generic symbolic ODEs. Do not provide biological meaning or biochemical details. Also specify if the network is valid or not.

Output

```
Nodes:  E2, S, SpE2, Sp, 2Sp, SE1, SSp, E1
Edges (Pixtral):  E2 → S (R1) E2 → Sp (R3) E2 → SpE2 (R4) S
→ E1 (R1) S → SSp (R5) Sp → 2Sp Sp → SE1 (R0) 2Sp → SE1
(R9) SE1 → E1 (R2) SSp → Sp (R6) E1 → SpE2 (R10)
Validity:  marked as DAG → valid.
ODEs (generic symbolic):  d[E2]/dt = f(E2) dS/dt = g(S,E2)
+ h(Sp) dSp/dt = i(SE1) + j(R6) dSSp/dt = k(S,E2) + l(Sp)
d2Sp/dt = m(Sp) dSE1/dt = n(Sp) + o(2Sp) dE1/dt = p(E1) +
q(SE1)
```

9.5 Key Observations

Across all three networks, several important differences emerge between the two models. First, ChatGPT shows a high degree of prompt sensitivity: in many cases it required carefully adjusted system prompts or rephrased user prompts to produce accurate graph interpretations, whereas the fine-tuned Pixtral model produced consistent outputs with the standard prompt alone. A related issue is model switching. Because `gpt-5-mini` refused to process Network 3 due to safety restrictions, a different ChatGPT model (`gpt-4o-mini`) had to be used, resulting in inconsistencies across the evaluation.

Another noticeable difference concerns validity assessment. ChatGPT incorrectly classified Network 1 as valid despite its structural ambiguity (specifically, an inhibitory edge targeting ϕ), while Pixtral correctly identified that the network was invalid. In terms of node and edge extraction, Pixtral adheres more closely to the literal features of the diagram, whereas ChatGPT frequently introduces interpretive assumptions or resolves graphical ambiguity by proposing alternative structural interpretations.

However, both models struggle with more complex diagrams. In the case of Network 3, both ChatGPT and Pixtral produced incorrect edge interpretations, which naturally propagated into incorrect ODE representations. This highlights the difficulty both models face when dealing with high-density or visually complex regulatory diagrams.

9.6 Conclusion

Overall, the fine-tuned Pixtral-12B model demonstrates **higher structural accuracy, lower prompt sensitivity, and more consistent extraction** across the three evaluated diagrams. While ChatGPT is more sensitive to prompt phrasing, prone to over-interpretation, and occasionally restricted by model safety filters. These results underscore the value of targeted fine-tuning for domain-specific diagram analysis tasks.

Use of AI Tools

AI-based assistance tools, including ChatGPT and Google Gemini, were used throughout the course of this project to support development, debugging, literature review, and documentation. These tools played a valuable auxiliary role in improving workflow efficiency and helping clarify complex implementation challenges. However, all core contributions of the project, including dataset construction, mathematical modeling, architectural design, fine-tuning experiments, validation procedures, and deployment, were performed manually and subjected to independent verification.

ChatGPT was used as a development assistant during model fine-tuning and pipeline construction. It helped explain parameter-efficient training techniques such as LoRA, assisted in debugging token alignment and formatting issues arising from multimodal prompt templates, and provided guidance when resolving compatibility conflicts involving PyTorch, PEFT, PIL image preprocessing, and the `transformers` [19] library. The tool was also consulted to refine JSONL dataset formatting and verify the logical consistency of annotation schemas required for supervision. During report preparation, ChatGPT assisted in editing and restructuring technical explanations to improve clarity, readability, and coherence without altering scientific content.

Google Gemini [20] was used primarily as a cross-verification resource. It assisted in independently checking symbolic expressions, including ODEs and Jacobians, and provided complementary perspectives during methodological decision-making related to training schedules, hyperparameter selection, and architectural trade-offs. This cross-consultation helped validate conclusions and ensured consistency in technical reasoning. The model also noticed to help better with the concepts of fine-tuning relating configurations and helped in figure out testing the bugs faced throughout the project.

Despite these advantages, several limitations were observed. AI-generated suggestions occasionally lacked domain-specific depth or introduced generic explanations when more nuanced biochemical or numerical interpretations were required. Generated code examples sometimes included inefficiencies or minor logical errors that required correction. Therefore, all AI-assisted guidance was treated as advisory only, and no generated content was incorporated into the dataset, training process, or final analysis without careful manual validation against established literature or direct mathematical verification.

The effective use of AI tools thus relied on maintaining critical human oversight and independent judgment to prevent inaccuracies or over-reliance on automated suggestions. When applied responsibly, these tools significantly enhanced productivity in debugging, technical writing, and conceptual clarification while preserving scientific rigor.

Overall, AI tools supported the research process by accelerating routine development tasks and facilitating clearer communication, but the intellectual ownership and scientific responsibility of all results contained within this report remain fully human-directed.

Challenges and Problems Solved

The development of a multimodal pipeline for translating biological network diagrams into structured mathematical models involved multiple technical, computational, and methodological challenges. Addressing these issues required iterative refinement across dataset development, model selection, preprocessing, and training design. This chapter summarizes the primary difficulties encountered and the approaches adopted to resolve them.

11.1 Dataset Creation Difficulties

A major early challenge was the absence of any publicly available dataset compatible with the project objectives. Existing biochemical resources lacked diagram images, mixed heterogeneous network types, or did not provide complete symbolic annotations such as ODEs and Jacobians. Since this work focused specifically on 1-D reaction networks, none of the available datasets could be reused directly.

Consequently, a custom multimodal dataset had to be constructed from scratch. An initial set of over forty diagrams was drawn manually to establish biological realism and stylistic consistency. To expand dataset scale, a Python-based synthetic network generator was implemented, producing more than 240 additional diagrams paired with corresponding reactions, ODEs, Jacobians, and species definitions. All samples required manual verification to ensure alignment between visual structures and symbolic outputs.

Annotating accurate ODEs and Jacobians presented an additional challenge. Although symbolic libraries such as SymPy aided generation, extensive validation was necessary to ensure correctness. The relatively small dataset size also raised the risk of overfitting during fine-tuning. This was mitigated by employing parameter-efficient LoRA adaptation and extended training schedules to improve stability.

11.2 Computational Constraints and Model Selection

Model selection was strongly influenced by hardware limitations. Initial fine-tuning experiments with `Mistral-Small-3.2-24B-Instruct-2506` [15] proved infeasible due to memory demands and repeated out-of-memory failures during multimodal training. These limitations prompted a transition to Pixtral-12B, whose more compact architecture and vision-aligned design enabled stable fine-tuning under available GPU resources.

11.3 Ambiguity and Variability of Biological Diagrams

Biological diagrams exhibit substantial visual variability in arrow styles, layouts, labeling conventions, and regulatory notation. Early experiments revealed frequent misinterpretations, including confusion between overlapping arrows, mishandling of inhibition symbols, and erroneous treatment of decay nodes such as ϕ as active species. Increasing dataset diversity through synthetic layout variation and inclusion of intentionally invalid diagrams improved robustness and reduced hallucinated reactions.

11.4 Multimodal Tokenization and Training Stability

Multimodal training required precise token formatting to align text inputs with image embeddings. Minor deviations initially caused collapse behaviors such as repeated outputs or ignored visual input. These issues were resolved by standardizing message construction with Pixtral's `apply_chat_template` and applying masking so that loss was computed only over the assistant response. Careful validation of the custom data collator stabilized training.

11.5 Generalization Challenges

Generalization remained difficult for diagrams significantly deviating from training layouts, including rotated arrows or visually dense reaction clusters. Additional synthetic augmentation and layout distortion were introduced to improve representation diversity, leading to more consistent generalization on unseen networks.

Overall, these challenges highlight the complexity of multimodal diagram interpretation and underscore the importance of carefully curated datasets, efficient model selection, and rigorous training procedures in achieving reliable biochemical model extraction.

Future Work

The system developed in this project provides a foundation for multimodal interpretation of biochemical network diagrams, but it also opens several avenues for deeper exploration and expansion. The work completed so far establishes the feasibility of extracting structured mathematical models from 1-D biological networks; however, real-world biochemical systems are considerably more complex, and future work should aim to approximate this broader landscape.

12.1 Extension Beyond 1-D Networks

A major direction for future expansion lies in moving beyond the exclusive focus on 1-D reaction networks. While concentrating on 1-D structures allowed us to maintain consistency during dataset construction and fine-tuning, such networks represent only a small fraction of biological regulatory behavior. In practical applications, biochemical systems often involve multi-pathway interactions, hierarchical regulatory layers, and spatially embedded processes that cannot be captured through 1-D representations alone. Extending the model to 2-D networks would enable the interpretation of diagrams containing branching pathways, cross-communication motifs, and more intricate signal flows. Further generalizing to 3-D or multi-layered networks would allow the system to address regulatory cascades, nested feedback loops, and systems exhibiting network modularity. Such expansions would not only increase dataset diversity but also substantially improve the robustness and relevance of the fine-tuned model for real biological modeling tasks.

12.2 Fine-Tuning at Larger Scale

The fine-tuning performed in this project was necessarily constrained by limited computational resources. As a result, we relied on parameter-efficient approaches and a compact model variant. With access to larger GPU clusters, future work could attempt full-model fine-tuning of Pixtral, allowing all weights, including the vision encoder, to adapt more deeply to biological diagrams. Exploring even larger multimodal architectures such as

Pixtral-22B or Qwen-VL-32B may further enhance the system's ability to capture subtle diagrammatic cues and resolve ambiguities that remain challenging for smaller models. Distributed training strategies would also enable experimentation with higher batch sizes, more extensive parameter sweeps, and deeper training regimes, all of which may yield measurable improvements in accuracy and generalization.

12.3 Integration of Vision Encoder Fine-Tuning

In the current system, only the language components of Pixtral were adapted through LoRA, while the vision encoder remained frozen. Although this approach provided computational efficiency, several studies indicate that fine-tuning specific visual layers can significantly improve performance on specialized diagrammatic domains. Future extensions of this project may involve partially fine-tuning the visual backbone, adapting the multimodal projection layers, or applying lightweight adapters to the region-level attention blocks of the vision encoder. Such modifications could help the model better recognize non-standard arrowheads, complex inhibitory symbols, and the varied stylistic conventions used in hand-drawn or literature-derived diagrams.

12.4 Expansion of Dataset and Autogeneration

The dataset constructed for this work, while carefully curated, remains relatively limited in scale compared to typical multimodal training corpora. Future progress will depend heavily on expanding this dataset along several dimensions. A natural next step is to generate thousands of synthetic 1-D, 2-D, and eventually 3-D biological networks using automated generation pipelines enriched with more diverse layouts, topologies, and regulatory mechanisms. Additional diagrams can be incorporated from textbooks, journal articles, pathway repositories, and other biological databases, offering exposure to real-world stylistic variations. Complementing these sources with noisy, handwritten, or scanned diagrams would help improve robustness, particularly for use cases involving manual sketches. At the same time, automating the generation of ODEs, Jacobians, and associated metadata at scale will enable the efficient construction of large, high-quality multimodal datasets capable of supporting more extensive fine-tuning.

12.5 Enhanced Mathematical Modeling

Although the current pipeline is limited to mass-action kinetics, extending the mathematical scope of the system is an important goal for future work. Many biological processes are governed by non-linear or saturating dynamics such as Michaelis–Menten kinetics, Hill-type cooperative effects, or stochastic fluctuations. Incorporating support for these modeling paradigms would enormously broaden the types of systems that the model can interpret. In addition, the extraction of higher-level analytical information, such as bifurcation structure, steady-state behavior, or sensitivity analyses, would allow the system to serve not only as a parser of diagrams but as a more complete computational modeling tool.

12.6 Synthetic Biology Applications

Finally, the system has the potential to contribute meaningfully to the domain of synthetic biology. With additional development, it could support automated circuit design, predict functional motifs, or aid in the analysis of oscillators, switches, and bistable systems. The ability to transition seamlessly between visual schematics and mathematical representations would make the model a valuable tool for iterative design and hypothesis testing. As the system expands beyond 1-D networks and incorporates richer mathematical frameworks, its applications in synthetic biology will become increasingly significant.

Conclusion

This project presents a multimodal, fine-tuned Pixtral-based system capable of interpreting biological network diagrams and translating them into machine-readable, mathematically rigorous models. By constructing a domain-specific dataset of over 240 annotated diagrams, including both valid and invalid networks, we equipped Pixtral with the ability to extract species, reactions, regulatory edges, ODEs, and Jacobians directly from images.

The work demonstrates that multimodal LLMs can serve as a bridge between visual biological notation and symbolic systems biology. The system leverages advances in low-rank adaptation, multimodal instruction tuning, and synthetic dataset construction to achieve reliable extraction performance on diagrams that vary widely in structure, complexity, and visual style.

While the current approach successfully converts diagrams into structured models, several challenges remain, including diagram ambiguity, visual irregularities, and the limits of multimodal model generalization. The transition from Mistral to Pixtral highlights the need for efficient architectures tailored to scientific imagery. The progress made here opens new avenues for larger-scale fine-tuning, improved dataset diversity, and integration with computational modeling frameworks.

Overall, this project establishes the feasibility of automated biochemical model extraction using multimodal LLMs and lays the foundation for future extensions in automated scientific reasoning, synthetic biology, and computational systems biology.

Implementation Details

The multimodal transcription system was implemented in Python using the Hugging Face ecosystem for model development and deployment. All training and inference experiments were executed on Linux-based GPU servers utilizing mixed-precision computation for memory efficiency.

14.1 Data Handling

Biochemical network diagrams were stored as image files and paired with structured annotations in JSONL format. Each dataset record contained a multimodal user prompt referencing the image and a corresponding assistant response specifying reactions, ODEs, Jacobians, and metadata. Data loading and batching were handled using the `datasets` library together with a custom multimodal data collator that ensured consistent formatting of text and image inputs.

14.2 Model Training

Fine-tuning was performed with the Pixtral-12B model using Low-Rank Adaptation (LoRA) via the `peft` framework. Training utilized the Hugging Face `Trainer` API with completion-only supervision, gradient accumulation, and gradient checkpointing to accommodate GPU memory limitations. All computations were carried out in `bfloat16` precision. Adapter checkpoints and processor configurations were saved upon training completion for reproducible inference deployment.

14.3 Inference

During inference, unseen diagram images were provided alongside standardized extraction prompts. Inputs were processed using the trained Pixtral processor and evaluated using deterministic decoding without sampling. Generated outputs were returned directly without postprocessing-based correction or schema enforcement.

14.4 Web Integration

The fine-tuned model was deployed as a persistent backend inference service powering the BioAutoMyze web platform, accessible at `http://bioautomyze.iiitd.edu.in/`. User requests are processed in real time by the GPU-hosted inference engine, enabling interactive diagram analysis through a browser-based chatbot interface.

14.5 Code Repository

The complete implementation of this project, including all source code, datasets, and instructions for running the simulations, is available on GitHub: [21] `https://github.com/ygyashgoyal/BioAutoCon`

Bibliography

[1] M. community, "Pixtral-12b model card (huggingface)," https://huggingface.co/mistral-community/pixtral-12b, 2024, model card; accessed 2025-11-XX.

[2] IIIT Delhi, "Bioautomyze web application," http://bioautomyze.iiitd.edu.in/, 2025, accessed: 2025-12-02.

[3] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, and et al., "Learning transferable visual models from natural language supervision," *arXiv preprint arXiv:2103.00020*, 2021. [Online]. Available: https://arxiv.org/abs/2103.00020

[4] J. Li, D. Li, C. Xiong, and S. C. H. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *ICML / arXiv*, 2022. [Online]. Available: https://arxiv.org/abs/2201.12086

[5] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, and et al., "Flamingo: a visual language model for few-shot learning," in *NeurIPS*, 2022. [Online]. Available: https://arxiv.org/abs/2204.14198

[6] "Vision–language model," https://en.wikipedia.org/wiki/Vision-language_model, 2024, accessed: 2025-12-01.

[7] K. Lee, A. Singh, and et al., "Pix2struct: Screenshot parsing as pretraining for visual language understanding," 2022. [Online]. Available: https://arxiv.org/abs/2210.03347

[8] M. AI, "Announcing pixtral," *Mistral AI Technical Blog*, 2024. [Online]. Available: https://mistral.ai/news/pixtral

[9] R. Madhavan, R. Lyne, H. Kitano, A. Doi *et al.*, "Biopax: A community standard for pathway data sharing," *Nature Biotechnology*, vol. 23, no. 8, p. 1035, 2005. [Online]. Available: https://www.biopax.org/

[10] A. Funahashi, M. Morohashi, H. Kitano, and N. Tanimura, "Celldesigner: a graphical biological network editor and simulation platform," *BMC Systems Biology*, vol. 1, no. 1, p. 134, 2007. [Online]. Available: https://celldesigner.org/

[11] M. Feinberg, “Chemical reaction network structure and the stability of complex isothermal reactors—i. the deficiency zero and deficiency one theorems,” *Chemical Engineering Science*, vol. 42, no. 10, pp. 2229–2268, 1987. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0009250987800994

[12] C. F. Lopez, J. L. Muhlich, J. A. Bachman, and P. K. Sorger, “Programming biological models in python using pysb,” *PLoS Computational Biology*, vol. 9, no. 11, p. e1003280, 2013. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/23423320/

[13] L. A. Harris, J. Hogg, J.-J. Tapia *et al.*, “Bionetgen 2.2: Advances in rule-based modeling,” *Bioinformatics*, vol. 32, no. 21, pp. 3366–3368, 2016. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC5079481/

[14] S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer, “Copasi—a complex pathway simulator,” *Bioinformatics*, vol. 22, no. 24, pp. 3067–3074, 10 2006. [Online]. Available: https://doi.org/10.1093/bioinformatics/btl485

[15] M. AI, “Mistral-small-3.2-24b-instruct-2506 model card,” *HuggingFace Model Card*, 2025. [Online]. Available: https://huggingface.co/mistralai/Mistral-Small-3.2-24B-Instruct-2506

[16] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, and et al., “Sympy: symbolic computing in python,” *PeerJ Computer Science*, vol. 3, p. e103, 2017. [Online]. Available: https://dx.doi.org/10.7287/peerj.preprints.2083v3

[17] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Chen, and et al., “Lora: Low-rank adaptation of large language models,” *arXiv preprint arXiv:2106.09685*, 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

[18] OpenAI, “Models — openai platform documentation,” https://platform.openai.com/docs/models, 2025, accessed: 2025-12-02.

[19] T. Wolf, L. Debut, V. Sanh, J. Chaumond, and et al., “Transformers: State-of-the-art natural language processing,” https://huggingface.co/docs/transformers, 2020, library / documentation; accessed 2025-11-XX.

[20] Google, “Gemini models — google ai studio documentation,” https://ai.google.dev/gemini, 2025, accessed: 2025-12-02.

[21] Devansh Kumar, Yash Goyal, “Github repository,” https://github.com/ygyashgoyal/BioAutoMyze, 2025, source code.