Phytomorphic Surrogate Neural Network for Synthetic Maize L-system Parameter Optimization

William Bushell[1 2], Ian McQuillan[1 3]

[1]Department of Computer Science, University of Saskatchewan

[2]william.bushell@usask.ca

[3]ian.mcquillan@usask.ca

August 2025

Abstract

Plant phenotyping and agricultural optimization require efficient methods for analyzing morphological parameters from procedural models. L-systems provide powerful procedural modeling frameworks for plant development, but parameter optimization remains computationally expensive due to non-differentiable simulation processes and costly image-based comparison methods. Traditional approaches using genetic algorithms or direct parameter search are computationally expensive for real-time agricultural applications. This paper presents a novel phytomorphic surrogate neural network architecture designed to enable gradient-based optimization of L-system parameters for maize plant modeling. Our two-stage framework first trains a surrogate model to predict plant comparison costs derived from image-processing distance functions between real maize images and L-system simulations. The phytomorphic model incorporates three specialized modules: Structure Generation Network for parameter embedding that outputs a probabilistic synthetic plant coordinate system, Hungarian Assignment Network for deep analysis and structural matching at each growth stage, and Cost Aggregation Network for final cost prediction. Stage two employs optimizer neural networks trained via backpropagation on surrogate predictions to discover optimal L-system parameters for target plant phenotypes.

Experimental validation on data from a single maize plant demonstrates that straight-forward surrogate neural networks cannot be leveraged for parameter optimization or produce biologically meaningful plant models. In contrast, the phytomorphic surrogate model enables genuine gradient-based optimization, successfully discovering plausible L-system parameters that minimize image-based structural dissimilarity to real plant phenotypes. The phytomorphic surrogate model is naturally robust to boundary exploitation during optimization, in contrast to simpler surrogate models, which tend to produce unrealistic parameter values at the boundaries of the search space. Overall, the surrogate-based neural optimization approach provides computational and interpretability benefits for plant phenotyping, with the phytomorphic surrogate model enabling gradient-based optimization of L-system parameters.

1. Introduction

Accurately modeling plant structure is essential for understanding growth and phenotype in crops. I am addressing the challenge of optimizing the parameters of an L-system-based plant model to generate synthetic plant images that closely resemble real specimens. Gradient-based optimization, a powerful technique for parameter optimization, are traditionally unsuitable for these tasks as L-systems are discrete and non-differentiable. This creates a vast and complex parameter space where small changes can lead to abrupt changes, either minuscule or massive shifts in plant structure.

The solution explored in this paper is the introduction of a neural network-based approach that enables gradient based optimization of L-system parameters through differentiable backpropagation. This approach supports rapid and meaningful optimization, providing new opportunities for plant modeling and agricultural applications. The issue is that the L-system plant generation is non-differentiable, meaning it cannot be utilized in the neural network approach. Instead, a differentiable surrogate, or replacement, is required.

In this paper, I specifically explore the viability of using a surrogate neural network for this problem, since neural networks are differentiable, and suited for this type of prediction task. However, the discrete nature of the L-system generation requires attention to detail and spatial awareness. To overcome these limitations, I introduce a differentiable phytomorphic neural network surrogate that learns the relationships between L-system parameters and image similarity. I call this surrogate phytomorphic since its architecture and training are designed to explicitly capture the hierarchical and modular organization of plant structures, reflecting their biological principles such as branching, growth stages, and morphological variation. The idea is that this neural network will be able to optimize the L-system parameters utilizing differentiable processes, enabling gradient-based search for biologically plausible parameter sets.

2. Methods

2.1 Parameter Space

The optimization seeks to fit a set of model parameters, encompassing both scalar values and control points for parameterized functions. Below is the full parameter set, their types, and representations.

The thirteen parameters used for L-system plant generation are as follows:

1. MAX_PHYTOMERS, the number of phytomers (structural units) in the plant.
2. PLASTOCHRON, interval between the initiation of phytomers or successive leaves.
3. IntLen, scaling factor for internode length.
4. IntWid, scaling factor for internode width.
5. ExpIntRad, exponent that controls how internode radius changes along the plant axis.
6. LeafLen, scaling factor for leaf length.
7. LeafWid, scaling factor for leaf width
8. ExpLeafWId, exponent that controls how leaf width changes along the leaf blade.
9. PlantDownAng, angle that determines the downward orientation of the entire plant.
10. PlantRollAng, angle that determines the rotational tilt of the plant structure.
11. BrAngle, angle at which the branches emerge from the main stem.
12. LEAF_BEN_SCALE, scaling factor for leaf bending.
13. LEAF_TWIST_SCALE, scaling factor for leaf twisting.

All model parameters except for MAX_PHYTOMERS are scalar-valued and continuous, taking real values within their specified ranges; MAX_PHYTOMERS is discrete and represented as an integer. The parameter space is dominated by continuous scalar-valued parameters, each varying smoothly within biologically plausible ranges. As a result, the optimization problem is essentially continuous and high-dimensional, with nonlinear interactions among parameters that give rise to complex emergent plant structures.

2.2 L-System Plant Generation Process

L-systems, or Lindenmayer systems, provide a formal grammar for simulating plant development through recursive production rules. Plant structures are generated by iteratively applying a set of rules that encode growth patterns and morphological differentiation. Each rule specifies how symbols in a string are replaced, with the resulting string representing the plant at a given developmental stage.

Parameterization of the L-system is achieved by associating model parameters with rule execution and geometric transformations. The thirteen parameters described in Section 2.1 control the behaviours of the plant growth. During simulation, these parameters are input into the L-system, which generates a sequence of plant structures corresponding to different developmental stages.

For each parameter set, the generated plant structure is rendered as a three-dimensional coordinate system representing key morphological features. This synthetic plant is then projected

into two-dimensional images using standardized imaging protocols. The imaging workflow ensures consistent viewpoints, lighting, and scale, enabling direct comparison between simulated plants and empirical images captured from real maize specimens. Images are processed to extract structural features, which serve as input for subsequent surrogate modeling and optimization stages.

These plant simulations were implemented using the VLAB modelling environment, with plant structure generation performed by the L-System Plant Functional Generator (LPFG) software. VLAB provides an integrated platform for parameter management, model execution, visualization by turtle graphics. LPFG interprets the L-system grammar and renders the resulting plant structures in three dimensions.

2.3 Baseline Surrogate Model Architectures

To benchmark the effectiveness of the proposed phytomorphic surrogate neural network, two baseline surrogate models were implemented and evaluated. These models serve as reference points, enabling comparative analysis of surrogate-driven optimization strategies. By systematically comparing these architectures to the phytomorphic approach, we assess the impact of surrogate design choices on prediction accuracy, optimization performance, and biological plausibility of discovered plant models.

Simple Surrogate Model:

The simple surrogate model employs a three-layer feedforward neural network to approximate the mapping from L-system parameter vectors to a scalar cost value representing plant structure similarity. The network receives the full set of thirteen normalized model parameters as input and outputs a single predicted cost. Training proceeds in an online fashion: after each new sample (consisting of randomly generated parameters and their corresponding simulation-based cost), the model weights are immediately updated via backpropagation using mean squared error loss. This one-at-a-time training strategy enables the model to rapidly adapt to new data. The surrogate is purely data-driven and non-modular, with no explicit representation of internal plant structure or hierarchical parameter effects.

Batch Training Surrogate Model:

The batch training surrogate model shares the same neural network architecture as the simple surrogate: a feedforward network mapping normalized L-system parameters to a predicted cost value. The key distinction lies in its training procedure. Here, samples are accumulated in batches of 16 before the model weights are updated. After a batch is filled, the model undergoes one training step in which gradients are averaged across all samples in the batch, followed by a single optimizer update. This approach can improve training stability and robustness to outliers, as the effect of individual noisy samples is mitigated through batch aggregation. Like the simple surrogate, the batch model remains non-hierarchical and non-interpretable, lacking explicit structural awareness or multi-stage modeling capabilities.

2.4 Two-Stage Training and Optimization Framework

The overall optimization procedure is organized into two sequential stages: surrogate model construction and surrogate-guided parameter optimization. This framework is designed to efficiently navigate the high-dimensional L-system parameter space while maintaining strong correspondence between simulated and empirical plant morphologies.

2.4.1 Stage 1: Surrogate Model Training

In the initial stage, a surrogate neural network is trained to approximate the mapping from L-system parameters to a quantitative plant structure similarity cost. Training data are generated by random sampling of parameter vectors within biologically plausible bounds, followed by procedural plant simulation and evaluation using image-based distance metrics.

This two-stage framework combines structurally aware surrogate modeling with efficient gradient-based optimization, offering substantial advantages in biological fidelity over traditional black-box or purely data-driven approaches.

Training Protocol:

All surrogate models utilize normalization of both input parameters and cost outputs, with normalization statistics (mean and standard deviation) computed from a representative set of L-system simulation outputs. During training, parameter vectors are normalized prior to input, and cost values are normalized as targets for regression. Model fitting is performed using supervised learning, where the objective is to minimize the mean squared error (MSE) between predicted and true normalized costs. For the simple and batch surrogate models, training proceeds either online (single-sample updates) or in batches (multi-sample gradient averaging), as outlined in Sections 2.3.1 and 2.3.2. Validation is performed by evaluating prediction error on held-out parameter-cost pairs not used during training.

Phytomorphic Surrogate Model Specifics:

During surrogate model training, the phytomorphic neural network incorporates both normalized L-system parameters and empirical plant feature coordinates across multiple developmental stages. These empirical features, extracted from manually annotated images of real maize plants over a 27-day growth period, include the spatial positions of key morphological landmarks such as leaf tips, stem nodes, and branching points at different time points. By providing this dual input, synthetic model parameters and temporally indexed real-world structural data, the surrogate network learns to map parameter configurations not only to static plant forms, but also to growth trajectories that reflect realistic development. Specialized loss terms are applied during training to penalize discrepancies in landmark assignments and spatial arrangements throughout the growth sequence, guiding the network to produce predictions that are both structurally faithful and temporally consistent with true biological development.

2.4.2 Stage 2: Surrogate-Guided Parameter Optimization

Following surrogate model training, the second stage employs a neural optimizer network to discover L-system parameter sets that minimize the surrogate-predicted cost for a target plant phenotype. This approach leverages the differentiability of the surrogate model, enabling direct

gradient-based optimization in a domain where the underlying simulation process is non-differentiable.

Optimization Process:

Optimization is performed via a custom neural network (OptimizerNet), which maps sampled latent vectors to candidate L-system parameter sets. The optimizer network consists of a sequence of fully connected layers with non-linear activations (ReLU) and a final Sigmoid activation, ensuring outputs are smoothly constrained within normalized parameter bounds. Each output is scaled to the biologically valid range for each parameter.

For each surrogate model, the optimizer network is trained using stochastic gradient descent with the Adam optimizer, performing multiple optimization restarts to increase robustness against local minima. The network takes random input so it can generate a wide range of candidate parameter sets, optimizing them in parallel, encouraging exploration, and ultimately improving the chance of finding high-quality solutions for your plant model fitting task.

A three-part composite loss function is employed to guide optimization:

- **Surrogate Cost Loss:** The primary objective is the mean predicted cost from the surrogate model, evaluated for the current batch of parameters.
- **Boundary Penalty:** A soft penalty term is added for parameter values that exceed biologically plausible bounds, calculated as the squared deviation from allowed parameter minima and maxima. This discourages solutions that exploit parameter space boundaries.
- **Diversity Loss:** An auxiliary loss term encourages variance among parameters in the batch, helping avoid premature convergence and promoting exploration.

The total loss is the weighted sum of these components, and gradients are propagated through the optimizer network and surrogate model via backpropagation. Learning rate scheduling is applied to facilitate convergence, and early stopping is triggered if no improvement in the loss is observed over a set number of iterations.

Multi-Restart Strategy and Model Selection

To further enhance solution quality, the optimization procedure includes multiple independent restarts for each surrogate model. Each restart initializes the optimizer network anew, with the best-performing parameter set (lowest surrogate-predicted cost) retained across runs. The optimizer network state corresponding to the best solution is saved for reproducibility.

Validation and Biological Plausibility

Upon completion of optimization for each surrogate model, the discovered parameter sets are exported and used to generate synthetic plant structures. These synthetic plants are quantitatively compared to real plant data using external cost metrics. Additional checks, including boundary constraint enforcement and plausibility regularization, are performed to ensure that optimized parameter sets yield biologically realistic plant morphologies, not merely cost-minimizing artifacts.

3. Phytomorphic Surrogate Model

3.1 Motivation and Design Philosophy

Procedural plant modeling, particularly with L-system parameterizations, presents unique challenges for surrogate-assisted optimization. Conventional surrogate models often lack structural awareness, treating parameter spaces as abstract vectors and ignoring the hierarchical, spatial, and developmental relationships inherent in plant phenotypes. This leads to several key limitations: surrogates may propose biologically implausible parameter values, are prone to boundary exploitation, and offer poor interpretability.

The phytomorphic surrogate model is designed to address these shortcomings through the following principles:

Structural Awareness and Biological Interpretability:

The model architecture reflects key aspects of plant development, incorporating modules that correspond to distinct biological processes: structure generation, assignment, and cost aggregation. This modular organization enhances interpretability and allows the model to capture hierarchical and spatial relationships within plant forms. By explicitly modeling plant structure, the surrogate improves robustness to boundary exploitation and unrealistic solutions.

Differentiability and Optimization Capability:

All modules of the model are constructed to be differentiable, supporting gradient-based optimization techniques such as backpropagation. This enables efficient parameter discovery and facilitates navigation through the high-dimensional, nonlinear parameter space typical of L-system models.

Modularity and Extensibility:

The separation of the model into specialized components supports ablation studies and future extensions, such as adaptation to additional plant species or integration with further data modalities. Constraints and regularization are incorporated to maintain optimization within biologically meaningful bounds.

The phytomorphic surrogate model is motivated by the need for optimization-aware surrogates that respect biological structure and deliver robust performance in complex plant modeling tasks.

3.2 Overall Architecture

The surrogate model is a modular neural network that transforms procedural plant parameters into a plant structure similarity cost. Its architecture is designed for interpretability, extensibility, and end-to-end differentiability, supporting robust training and optimization. The network consists of sequential submodules, structure generation, assignment, and cost aggregation, each passing formatted outputs to the next, enabling clear data flow and gradient propagation throughout the system.

Modules use neural network components suited to their data, such as fully connected layers for parameter transformations and attention mechanisms for feature matching. Standard deep learning practices, including activation functions, normalization, and dropout, are employed to ensure generalization and stability. Training is end-to-end via gradient-based optimization, with similarity cost as the primary loss. The modular design facilitates component replacement without disrupting overall performance.

Implementation with modern deep learning frameworks ensures scalability, reproducibility, and integration with additional data modalities. Throughout, the model maintains clarity in data interfacing and consistent input/output handling, supporting both interpretability and future development. The following subsections describe each module in detail.

3.2.1 Structure Generation Network

The structure generation module constitutes the initial stage of the surrogate model pipeline. Its role is to transform the procedural plant parameters, which are encoded as a fixed-length vector according to the L-system formalism, into a geometric representation of plant morphology. Specifically, these parameters define the production rules, recursion depth, and segment lengths that would conventionally be used for explicit simulation of branching structures.

In this implementation the module is realized as a multilayer perceptron. The input is a vector containing the L-system parameters, and this is processed by a sequence of fully connected layers. Each hidden layer applies a linear transformation followed by a rectified linear unit (ReLU) activation function. Batch normalization is performed after each hidden layer to promote stable and efficient training dynamics. The output of the structure generation module is a set of two-dimensional coordinates representing the predicted locations of branch points and endpoints. These coordinates are standardized to a common reference frame so that they can be compared directly with reference plant structures in subsequent stages of the model. The module supports a variable number of predicted points by masking unused outputs and standardizing the active predictions. This masking is implemented by assigning a specific value to unused outputs and ensuring downstream modules ignore these during assignment and cost computation.

During training, supervision is provided by comparing the predicted feature coordinates to empirically observed positions in a dataset of real or simulated plant structures. Gradients are propagated from the final cost output back through the assignment and cost aggregation modules, allowing the structure generation module to optimize its parameterization for accurate similarity prediction. This approach enables the surrogate model to rapidly infer plant morphology directly from procedural parameters, supporting efficient end-to-end optimization. The design of this module abstracts the process of procedural structure generation, embedding it within a differentiable neural network framework. This allows the entire surrogate model to remain compatible with gradient-based optimization and facilitates integration with subsequent modules for assignment and cost aggregation.

3.2.2 Hungarian Assignment Network

The Hungarian Assignment Network is responsible for establishing optimal correspondences between predicted morphological features produced by the Structure Generation Network and reference features derived from empirical maize plant data. Its primary function is to enable a deep, quantitative comparison between simulated and real plant structures across all developmental stages.

At each timepoint, the network receives sets of predicted and reference features, such as branch points or organ positions, and constructs a cost matrix where each entry encodes the Euclidean distance between a predicted feature and a reference feature. The core assignment step utilizes the Hungarian algorithm, which deterministically produces the globally optimal one-to-one pairing that minimizes the total assignment cost. Assignments arising from padded entries are systematically ignored, so only valid pairings contribute to the final evaluation. The output of the Hungarian Assignment Network is a set of matchings that represent the closest possible correspondence between simulated and real morphological features for each growth stage. These assignments are used to aggregate costs, yielding a similarity measure that serves as both an interpretable evaluation metric and a differentiable loss for neural network training. By structuring the assignment process to be compatible with gradient-based optimization, the module enables efficient learning and parameter discovery throughout the surrogate modeling pipeline.

Through its design, the Hungarian Assignment Network supports robust and efficient comparison of plant structure and dynamics, ensures deterministic and interpretable assignments, and provides critical structural information for downstream cost aggregation and optimization. This approach facilitates biologically meaningful parameter optimization and advances the overall fidelity and reliability of the phytomorphic surrogate neural network framework.

3.2.3 Cost Aggregation Network

The cost aggregation network is responsible for converting a sequence of daily assignment costs, each representing the difference between synthetic and real plant structures on a given day, into a single value that summarizes the overall cost or fitness of a plant. This network enables the model to reason about plant performance over time in a way that is both learnable and efficient.

Rather than using a simple sum or average, the network uses a stack of fully connected layers to learn how to combine the daily costs into a final scalar output. Specifically, the network transforms the input through a sequence of nonlinear operations: each layer learns weighted combinations of the daily values, and the nonlinearity (ReLU) allows the network to capture interactions between days and learn which days are most important for predicting overall cost.

This approach allows the aggregation to be flexible: the network can learn to give more weight to critical time points or combine temporal patterns in complex ways if that improves prediction. The final output is a single number that represents the plant’s predicted cost, which is used for training and evaluation.

The main advantage of this design is that it does not assume any particular rule for how costs across days should be aggregated. Instead, it lets the model discover an effective strategy based on the data and the training objective. This is especially important in biological systems, where the importance of a particular day or developmental stage is not always known in advance and may depend on subtle interactions in the phenotype's trajectory.

Because the network is fully connected and not recurrent, it treats all days equally at the start of training but can learn to specialize as optimization proceeds. The model is simple and fast to train but could potentially be extended with temporal models if more complex patterns in the daily costs are found to be relevant. By using a learnable nonlinear transformation, the network can adaptively weight and combine costs from different days, giving more influence to days that are most predictive of plant fitness and modeling interactions between days if useful. This approach allows the model to discover the most effective aggregation strategy based on training data, rather than relying on manually defined rules.

4. Experimental Setup

4.1 Overview

This section details the experimental framework used to evaluate the proposed phytomorphic surrogate model and competing baseline approaches. The experiments are designed to assess model prediction accuracy, robustness against boundary exploitation, and effectiveness in parameter optimization tasks relevant to structural plant phenotyping.

Synthetic plant structures for all experiments were generated using the LPFG L-system simulator and the University of Calgary's VLAB platform, employing maize-specific grammars to ensure biological realism.

We compare the phytomorphic surrogate against established surrogate architectures across multiple criteria, including cost prediction fidelity and optimization performance. Special emphasis is placed on analyzing the tendency of surrogate models to exploit parameter space boundaries, as well as the ability of the phytomorphic model to generate biologically plausible plant structures. Experimental protocols, evaluation metrics, and implementation details are described to ensure reproducibility and facilitate meaningful comparison. The subsequent sections specify the dataset, model training configurations, and the methodology used for quantitative analysis.

4.2 Experimental Design

The experimental setup enables direct comparison between multiple surrogate neural network models for hierarchical plant phenotyping. Each surrogate model is trained independently using a set of randomly generated samples, drawn from a consistent parameter distribution. Datasets for each model are generated separately, so while the sampling protocol is identical, the specific training samples vary between models.

For each evaluation sample, a real plant structure is generated via VLAB based on the sampled parameters. The true cost is then computed by quantitatively comparing this simulated maize plant structure to actual measured plant data using a domain-specific cost function. The surrogate model's predicted cost for the same parameter set is directly compared to this real cost, providing a measure of prediction fidelity.

After training, each model is evaluated on a newly generated batch of random samples from the same underlying distribution. For each sample, both the predicted cost (from the model) and the true cost are computed and recorded in standardized CSV files. The evaluate_model.py script processes these results, automatically scanning the output directory for CSV files from different models.

To ensure fair comparison, the minimum sample count across all result files is used for each model. Samples are grouped into 100 bins for robust averaging and visualization. For each model, key performance metrics are computed for the full set of groups (across all 100 bins), as well as specifically for the final bin. This allows assessment of both overall performance and end-state

behavior. Where available, additional metrics such as cost loss, total loss, and parameter sensitivity are included.

The assessment protocol generates comprehensive visualizations (cost loss evolution, relative error evolution, accuracy curves, predicted versus true cost scatter plots, error distributions, and comparative bar plots for final performance) and statistical summaries (mean, median, and standard deviation of relative error, R^2 score, and convergence stability) for each model. These statistical summaries are primarily calculated using data from the final bin.

All evaluation steps are applied uniformly to every model, ensuring quantitative and qualitative assessment of surrogate architectures under equivalent experimental conditions.

4.3 Evaluation Metrics

The following metrics are used to assess surrogate cost prediction performance where N is the number of evaluated samples:

- **Average Predicted Cost**
  Mean value of surrogate model cost predictions across evaluation samples.
  Formula:
  $$\text{Average Predicted Cost} = (1/N) \times \Sigma \text{ (Predicted Cost for each sample)}$$

- **Average True Cost**
  Mean value of ground truth costs for the same evaluation samples.
  Formula:
  $$\text{Average True Cost} = (1/N) \times \Sigma \text{ (True Cost for each sample)}$$

- **Average Relative Error**
  Mean absolute relative error between predicted and true costs.
  Formula:
  $$\text{Average Relative Error} = (1/N) \times \Sigma \text{ |(Predicted Cost – True Cost) / True Cost|}$$

- **Accuracy Thresholds**
  Fraction of samples with relative error below specified thresholds:
  - Accuracy < 1%: Proportion where Relative Error < 0.01
  - Accuracy < 5%: Proportion where Relative Error < 0.05
  - Accuracy < 10%: Proportion where Relative Error < 0.10

- **R^2 Score (Coefficient of Determination)**
  Measures the proportion of variance in true cost explained by surrogate predictions.
  Formula:
  $$R^2 = 1 - [\Sigma(\text{Predicted Cost} - \text{True Cost})^2 / \Sigma(\text{True Cost} - \text{Average True Cost})^2]$$

- **Statistical Summaries of Relative Error**
  Mean, median, and standard deviation of the relative error distribution.

- **Convergence Stability**
  Standard deviation of average cost loss over the last ten evaluation bins, indicating prediction stability near convergence.

- **Parameter Sensitivity**
  Absolute Pearson correlation between each input parameter and true cost, where applicable.

All metrics are computed for evaluation samples grouped into bins, with special emphasis on final bin performance to assess end-state accuracy and stability.

4.4 Training Procedures and Hyperparameters

All surrogate models were implemented in Python using PyTorch. Training and evaluation were performed on randomly generated L-system parameter sets with corresponding cost values derived from plant structure comparisons. The following procedures and hyperparameters were used:

- **Model Architectures:**
  - The simple and batch surrogate models use a three-layer feedforward neural network with an input dimension of 13 (representing L-system parameters), two hidden layers of 64 units each with ReLU activation, and a single output neuron for cost prediction.
  - The hierarchical (phytomorphic) surrogate model consists of modular sub-networks for structure generation, Hungarian assignment, and cost aggregation. The input dimension is 13; hidden layers range from 64 to 256 units; the maximum number of plant structure points is set to 50; cost aggregation is performed over up to 27 days.
- **Normalization:**
  - Input parameters and cost outputs are normalized using statistics computed from 100 randomly sampled L-system simulations prior to training for each model.
- **Training Protocol:**
  - Each model is trained independently for *n* samples, as specified by the user.
  - The loss function is mean squared error (MSE) for cost prediction. The hierarchical model also includes regularization terms and a multi-component loss.
  - The Adam optimizer is used for all models.
  - The learning rate is 0.001 for the simple and batch surrogate models, and 0.0001 for the hierarchical surrogate model.
  - The batch size is 16 for batch models, while the simple surrogate model uses online training (weight updates after each sample). The hierarchical model also uses single-sample updates.
  - There is no explicit early stopping or learning rate decay; training runs for a fixed number of samples.
  - Model weights are saved to disk after training is complete.
- **Evaluation:**
  - At each training step, predictions and losses are logged to CSV files.
  - Metrics such as average loss, relative error, and accuracy at thresholds of less than 1%, 5%, and 10% are tracked during training.
  - Surrogate models are evaluated on held-out random samples for final performance comparison.

- **Other Details:**
    - Training progress, accuracy, and estimated time remaining are printed at each step.
    - Models that fail to converge or produce invalid cost predictions are excluded from analysis.

This standardized protocol ensures direct and reproducible comparison of surrogate architectures.

5. Results

This section presents the experimental findings in the two main domains of this paper: (1) parameter optimization for image-matching L-system maize models to real maize plant images, and (2) evaluation of the phytomorphic surrogate neural network approach for cost prediction and optimization robustness in comparison to the benchmark surrogate models.

5.1 Surrogate Cost Prediction Performance Analysis

This section assesses how accurately each of the three surrogate models predicts the image-based cost function given L-system maize parameters.

All surrogate models were trained and validated on 480,000 held-out samples, with key metrics including the average relative error (%), R^2 score, and accuracy thresholds. Table 5.1 summarizes the results found in the last 4,800 samples of training:

Model	Acc <1%	Acc <5%	Mean Rel Error	Median Rel Error	R^2 Score	Convergence Stability
Benchmark1	74.3%	99.7%	0.0077	0.0054	0.9911	0.000223
Benchmark2	67.4%	99.7%	0.0088	0.0066	0.9887	0.000162
Phytomorph	71.8%	99.7%	0.0081	0.0057	0.9896	0.000254

Table 5.1

All models exhibit strong performance, with Benchmark1 attaining the highest accuracy at the strictest threshold (<1%), the lowest mean and median relative errors, and the highest R^2 score. Benchmark2 follows closely, while the Phytomorphic surrogate maintains competitive accuracy, particularly at the <5% threshold, matching the benchmarks at 99.7%. Although the phytomorphic's accuracy at <1% is slightly below Benchmark1, it demonstrates robust convergence stability comparable to the benchmarks. These results provide a solid basis for further analysis of boundary exploitation and optimization robustness.

5.2 Surrogate Parameter Optimization Performance Analysis

This section assesses how each of the three surrogate models performs in the optimizer network. Notably, only the phytomorphic surrogate was successful in producing realistic and biologically plausible solutions; both benchmark surrogates (simple and batch) exhibited severe boundary exploitation, resulting in implausible parameter sets and cost values.

Each optimizer network trained with 30 restarts, with an early stopping functionality if a local minimum is found. Below is the best matching plant each surrogate can provide under the optimizer neural network, as well as the correct number of phytomers and the recorded number of phytomers.

Model	True Cost	Pred Cost	Real Phytomers	Predicted Phytomers

Benchmark1	59626	-65.7	9	8
Benchmark2	59543	0.4	9	8
Phytomorph	50097	52744	9	10

Table 5.2

As seen above, only the phytomorphic surrogate model functioned when being used in the optimizer network. The phytomorphic model was able to find parameters that led to an L-system generating with 50097 cost, while keeping a realistic predicted cost itself.

5.3 Boundary Exploitation: Investigation and Evidence

Comprehensive analysis of the optimized parameter sets generated by benchmark surrogate models reveals a pervasive boundary exploitation phenomenon. Across all 30 optimization restarts for each benchmark surrogate, 100% of the final parameters were located almost exactly at their minimum or maximum allowable values, with no exceptions. The consistent pattern strongly indicates that the cost landscapes shaped by the benchmark surrogates incentivize boundary solutions, rather than biologically plausible interior ones.

6. Discussion

6.1 Core Contribution: Neural Networks for Direct Parameter Optimization

This study establishes a method by which neural networks act as powerful surrogates for direct parameter optimization in complex, image-driven modeling tasks. By leveraging differentiable architectures, the approach enables efficient search for model parameters that yield realistic and biologically plausible outcomes, advancing both theory and practice in surrogate-based optimization.

6.1.1 Theoretical Innovation and Differentiability

A central theoretical contribution is the development of differentiable neural surrogates capable of emulating the cost landscape for L-system modelling and image matching. Unlike traditional surrogate models, which lack gradient information or structural flexibility, the proposed neural architectures facilitate gradient-based optimization. The phytomorphic surrogate extends this framework through structural awareness and modularity, introducing domain-specific inductive biases that improve the alignment between surrogate predictions and true process dynamics.

6.1.2 Empirical Effectiveness in Optimization Tasks

Empirical evaluations demonstrate that, while all surrogate models achieve high predictive accuracy on held-out cost data, only the phytomorphic surrogate consistently produces parameter sets that are both plausible and robust. Benchmark models (simple and batch) suffer from boundary exploitation, converging to degenerate solutions that lack biological realism. In contrast, the phytomorphic surrogate drives optimization toward interior solutions, maintaining meaningful coverage of the parameter space and realistic cost values. These results confirm the practical benefit of architectural innovations tailored to the underlying process.

6.1.3 Practical and Cross-Domain Applications

The approach outlined here is broadly applicable beyond the domain of plant morphology. Neural surrogates for direct parameter optimization can be adapted to diverse scientific and engineering contexts where complex generative models must be fit to real-world data, such as materials design, medical image analysis, or physics-based simulation. The demonstrated integration of differentiability, modularity, and domain awareness provides a generalizable blueprint for robust surrogate model development across disciplines.

6.2 Surrogate and Phytomorphic Model Architecture Insights

6.2.1 Comparative Performance Across Architectures

The comparative evaluation of surrogate architectures reveals clear distinctions in optimization outcomes. Standard surrogates demonstrate high predictive accuracy in cost estimation but consistently fail to produce biologically plausible parameter sets when used for derivative-based optimization. In contrast, the phytomorphic surrogate model integrates domain-specific structural

awareness, enabling it to generate robust, realistic solutions that avoid degenerate edge cases. These results underscore the importance of architectural design in shaping the quality and reliability of optimization processes.

6.2.2 Boundary Effects and Mitigation Strategies

A recurring challenge in surrogate-based optimization is boundary exploitation, where optimized parameter sets converge at the limits of their allowable ranges. Benchmark surrogates were particularly susceptible to this effect, resulting in implausible solutions and artificially low-cost values. The phytomorphic surrogate mitigates these issues through modularity and structural priors, which reshape the cost landscape to discourage boundary clustering. This architectural innovation demonstrates the value of embedding domain knowledge to enhance the robustness of neural networks and surrogate models.

6.2.3 Biological Realism and Optimization Advantages

Biologically inspired surrogate designs offer tangible advantages in both process emulation and optimization fidelity. The phytomorphic surrogate's incorporation of plant-specific structural features leads to parameter sets that more accurately reflect true biological variation, as evidenced by interior coverage of the search space and realistic plant morphologies. This improvement in biological realism translates directly into more meaningful and credible optimization outcomes.

6.2.4 Design Guidelines for Optimization-Oriented Surrogates

The findings suggest several actionable principles for future surrogate model development. First, incorporating domain knowledge and structural awareness is critical to prevent boundary exploitation and improve solution plausibility. Second, modular architectures support flexible adaptation to new problem contexts and facilitate integration of specialized inductive biases. Finally, rigorous validation against real-world data should accompany surrogate model design to ensure both predictive accuracy and biological realism.

6.2.5 Architectural Integration and Innovation

Neural surrogate frameworks provide a versatile foundation for integrating a wide range of architectural approaches. The success of the phytomorphic surrogate illustrates how modularity, differentiability, and domain-specific priors can be combined to advance both optimization and process emulation. Future innovation will likely involve deeper integration of neural and symbolic modeling techniques, enabling broader applicability and enhanced interpretability in complex scientific domains.

6.3 Limitations and Challenges

6.3.1 Persistent Boundary Exploitation Issues

Despite architectural advances, persistent boundary exploitation remains a noteworthy challenge in surrogate-based optimization. While the phytomorphic surrogate model substantially mitigates

clustering of parameters at the limits of their allowable ranges, certain optimization scenarios could still yield edge-convergent solutions, especially under extreme or poorly constrained search spaces. This underscores the need for further research into regularization strategies, landscape smoothing, and advanced constraints to ensure consistently plausible parameter distributions.

6.3.2 Generalization and Robustness Constraints

Another limitation involves the generalization capacity and robustness of surrogate models. The empirical results demonstrate high accuracy and biological realism within the tested dataset; however, it is unclear how well these models extrapolate to novel parameter regimes, unseen plant forms, or different environmental conditions. Overfitting to the training data, limited coverage of the morphological diversity, and sensitivity to noise or outliers may restrict broader applicability. Future work should include cross-validation with more diverse datasets and systematic robustness testing to better characterize generalization limits.

6.3.3 Temporal Growth Function in Phytomorphic Model

A specific challenge for the phytomorphic surrogate lies in the accurate modeling of temporal growth dynamics. While the current implementation emulates static plant morphologies effectively, it does not fully capture the time-evolving nature of plant development. Incorporating explicit temporal growth functions or spatiotemporal modeling frameworks could substantially enhance biological realism and predictive utility. This represents a key direction for future refinement and application, particularly for studies interested in developmental processes or longitudinal data.

7. Conclusion

This work introduces a novel framework for gradient-based optimization of procedural plant models, addressing longstanding challenges in parameter fitting for L-system representations of maize. By developing the phytomorphic surrogate neural network, we enable direct, differentiable optimization of high-dimensional, non-differentiable plant model parameters, replacing traditional genetic algorithms and direct search methods with efficient, interpretable neural surrogates. The modular architecture embeds biological priors and structural awareness, ensuring robust and realistic plant phenotype matching.

Empirical validation demonstrates two key advances. First, our surrogate-based optimization approach successfully discovers plausible L-system parameters that minimize structural dissimilarity to real maize phenotypes, overcoming the computational cost and boundary exploitation issues inherent in simpler surrogate models. Second, the phytomorphic surrogate model achieves biologically meaningful interior solutions, maintaining high predictive accuracy while reliably avoiding degenerate parameter sets at the boundaries of the search space.

Together, these contributions advance the capabilities of surrogate modeling for plant phenotyping and procedural model fitting, offering both computational efficiency and enhanced biological realism. The framework is extensible to broader agricultural optimization problems and other domains where complex generative models are used. Future work may explore dynamic growth modeling, generalization to other species, and integration of environmental variables, further expanding the scope and impact of this approach.

7.1 Acknowledgments

Acknowledgements:

1. Supported by an NSERC-USRA for April-August 2025
2. Research conducted as part of the Accelerated Master’s program at the University of Saskatchewan
3. Supervised by Professor Ian McQuillan, Department of Computer Science, University of Saskatchewan
4. L-system provided by the Biological Modeling and Visualization Laboratory at University of Calgary
5. Supporting software provided by Nazifa Khan, PhD candidate

7.2 References

1. Prusinkiewicz, Przemyslaw, and Aristid Lindenmayer. The Algorithmic Beauty of Plants. Springer New York, 1990. The Virtual Laboratory.

7.3 Codebase

All source code for the phytomorphic surrogate neural network, baseline surrogate models, training scripts, and evaluation procedures is available at:

https://github.com/bushwill/Surrogate-Neural-Network

The repository includes:

- Implementation of the hierarchical phytomorphic surrogate neural network architecture.
- Scripts for L-system plant simulation, data preprocessing, and cost calculation.
- Training and evaluation workflows for all models discussed in this paper.

Access to the code ensures full reproducibility of the results presented and enables extension of the methods to new datasets or plant species.