A thesis submitted in partial satisfaction of the requirements for the degree of Master of Computer Science and Engineering in the Graduate School of the University of Aizu

Flocking with Cellular Automata: Exploring the Intersection of Boid Models and Lenia Dynamics

by

Ryoma Usui

March 2025

The thesis titled

Flocking with Cellular Automata:
Exploring the Intersection of Boid Models and
Lenia Dynamics

by

Ryoma Usui

is reviewed and approved by

March 2025

Chief referee

Senior Associate Professor Date:

Maxim Mozgovoy

Professor Date:

Rentaro Yoshioka

Associate Professor Date:

Yutaka Watanobe

THE UNIVERSITY OF AIZU

Contents

List of Figures and Tables

List of Abbreviations

CA Cellular Automata
ALife Artificial Life
GoL Conway's Game of Life
EA Evolutionary Algorithm
KDE Kernel Density Estimation
SLPs Spatially Localized Patterns

List of Symbols

$\mathbf{x}_i$	The position vector of point i in a 2D space
$\boldsymbol{\mu_k}$	A parameter to adjust the shape of the kernel
$\boldsymbol{\sigma_k}$	A parameter that defines the center and dispersion of the growth mapping
$\boldsymbol{h_k}$	A parameter to adjust the weight of elements within the kernel
$\mathbf{G}(\mathbf{x})$	Gradient or force vector at position $\mathbf{x}$
$\boldsymbol{c_{\text{sep}}}, \boldsymbol{c_{\text{align}}}, \boldsymbol{c_{\text{cohe}}}$	The strengths of separation, alignment, and cohesion in the boid model, respectively
$\boldsymbol{d_{\text{sep}}}, \boldsymbol{d_{\text{align}}}, \boldsymbol{d_{\text{cohe}}}$	The attenuation strengths of separation, alignment, and cohesion due to overlap in the boid model
$\mathbf{p}_i$	Position of the boid in 2D space
$\mathbf{v}_i$	Velocity vector of the boid in 2D space
$\boldsymbol{v_{\text{max}}}$	Maximum allowable speed for Boids
$\boldsymbol{\gamma}$	Velocity damping (scaling) coefficient for Boids
$\boldsymbol{r_k}$	Radius parameter for the Lenia kernel
$\boldsymbol{a_i}$	distance from paritlcle of kernel bump
$\boldsymbol{w_i}$	width of kernel bump
$\boldsymbol{b_i}$	height of kernel bump
$\boldsymbol{K(r)}$	Kernel value at radius r
$\boldsymbol{N}$	Number of Boids (or particles) in the simulation
$\boldsymbol{E_{\text{flock}}}$	Total potential energy in the system
$\boldsymbol{V_{\text{flock}}}$	Total kinetic energy in the system
$\boldsymbol{t}$	Discrete time index or continuous time variable in the simulation

Acknowledgement

I would like to express my sincere gratitude to my supervisor, Prof. Maxim Mozgovoy, for their guidance and support throughout my research. His advice has been invaluable in completing this thesis.

I would also like to thank my friends and family for their encouragement and understanding during my studies.

Abstract

Understanding the complexity of dynamic systems through simulation is a critical theme in the study of artificial life and swarm behavior. In this research, we propose a novel dynamics model that integrates the “Boid model,” which reproduces swarm behavior through simple interaction rules, with “Lenia,” which leverages nonlinear dynamics to create life-like patterns. This model simulates the interaction between swarm behavior and cellular dynamics.

Specifically, we constructed a system in which Lenia’s growth field is incorporated into the Boid model, allowing Boid agents to influence Lenia particles while Lenia’s dynamics, in turn, affect the movement of the Boid agents. This integrated model enables Lenia’s growth rate to regulate the cohesion and collective motion of Boids while Boid movement contributes to Lenia’s pattern formation.

The simulation results revealed new dynamic behaviors. Notably, enhanced aggregation of Boids in regions with high Lenia growth fields and the formation of asymmetric growth patterns were observed. These findings suggest the possibility of reproducing complex and organic swarm behaviors and pattern formations that conventional models could not achieve.

The outcomes of this research provide a novel framework linking swarm intelligence and self-organizing systems, paving the way for diverse applications in artificial life, robotics, and understanding biological systems.

Chapter 1

Introduction

Modern simulations of artificial life and swarm behavior have become essential tools for understanding complex dynamic systems. Notably, the Boid [1] model and simulations like SmoothLife [2] and Lenia [3] have garnered attention for mimicking swarm behavior and cellular dynamics through distinct approaches. The Boid model is widely used for dynamic simulations of collective movements in swarms or flocks, generating coordinated behavior through simple rules governing interactions between individuals. In contrast, Lenia leverages nonlinear dynamics to simulate cellular movements, capturing life-like behaviors that transcend physical constraints.

While each of these models is fascinating in its own right, addressing their limitations and creating more realistic and intricate dynamic systems requires a novel perspective beyond treating them in isolation.

1.1 Research Objectives

The objective of this research is to propose a novel dynamic model that integrates the Boid model and Lenia simulation, facilitating interactions between swarm behavior and cellular dynamics. Through this integration, the collective behavior in the Boid model will influence the dynamics of Lenia particles, while Lenia particles, in turn, will affect individual agents in the Boid system. Specifically, the Lenia particles will exert cohesion and alignment forces on Boid agents. To achieve a more natural interaction, the concept of growth will be introduced as a parameter in these interactions.

1.2 Significance of the Research

The significance of this research lies in uniting two seemingly distinct fields—swarm behavior and cellular dynamics—into a cohesive framework for modeling more complex and realistic dynamic systems. This integrated approach has the potential to provide fresh perspectives in artificial life, robotics, swarm simulations, and even the understanding of biological systems.

Chapter 2

Background

We here introduce the foundational concepts of two prominent models: the Boid model, which simulates swarm dynamics, and Lenia, which captures life-like patterns through continuous cellular automata.

2.1 Boid

The Boid model [1] is an algorithm used to simulate the movement of swarms or flocks, based on three fundamental principles: "Separation," "Alignment," and "Cohesion." These rules determine how individual agents (Boids) interact with one another.

- **Separation**: This rule generates a force that causes a Boid to avoid others if they get too close. This helps prevent collisions among Boids within the group.
- **Alignment**: This rule encourages a Boid to align its direction with nearby Boids, promoting harmony within the swarm's overall movement.
- **Cohesion**: This rule generates a force that pulls a Boid toward the center of the swarm, ensuring that the group moves as a cohesive unit.

These rules are based on local interactions between individuals, yet they collectively produce organized behavior within the swarm. This model is widely used to replicate collective behaviors observed in nature, such as flocking birds, schooling fish, and swarming insects.

2.2 Lenia

Lenia is a continuous cellular automaton that extends traditional cellular automata by using continuous space, time, and states [3]. This enables complex, life-like patterns that are not achievable with discrete cellular automata. Lenia achieves this by employing multiple kernels and channels, which drive self-organizing dynamics.

Lenia's system is defined in a d-dimensional continuous space, though most implementations use a 2-dimensional plane ($d = 2$). Each point in this space contains c real-valued channels, with values ranging between 0 and 1. The state of the system evolves over time based on local interactions, which are defined by kernels and growth mappings.

Lenia is innovative in its use of continuous space, time, and states, setting it apart from traditional cellular automata. This allows Lenia to simulate complex and realistic life-like behaviors that were previously unattainable with conventional cellular automata. Lenia utilizes multiple kernels and multiple channels to generate self-organizing, autonomous patterns.

The world of Lenia begins with a space defined as a d-dimensional lattice in its initial configuration. Each lattice point contains c channels, represented by real values between 0 and 1. The state of each cell evolves over time based on these channels.

Lenia's evolution is driven by **multiple kernels** and **growth mappings**. The evolution process is computed as follows:

- **Kernel**
  Each kernel serves as a function that describes the interactions between cells. Kernels are characterized by features such as radius r_k, parameter β_k, growth mapping parameter μ_k, and σ_k. A kernel is constructed by combining exponential kernels and kernel shells to describe the spatial influence of cells.
- **Growth Mapping**
  Growth mappings are functions used to update the states of cells, employing distinct parameters for each kernel. Growth mapping can be defined, for example, as follows:

$$G_{k(u)} = 2 \cdot \exp\left(-\frac{1}{2}\left(\frac{u-\mu_k}{\sigma_k}\right)^2\right) - 1$$

Here, u represents the state of the cell, while μ_k and σ_k are parameters indicating the center and spread of growth, respectively.

2.2.1 Update Calculation

In Lenia, the state of the world is updated based on the following equation:

$$A_j^{t+\Delta t} = \left[A_j^t + \Delta t \sum_{i,k} \frac{h_k}{h} \cdot G_k(K_k * A_i^t)\right]_0^1$$

In this equation, the effects of each kernel are averaged, and the **weighted composite result** is applied as the new state. Kernels interact based on the states of surrounding cells, determining the evolution of the cells.

Each kernel is defined by the following parameters:

- r_k (Radius): A parameter that determines the range of influence of the kernel.
- β_k: A parameter that adjusts the shape of the kernel.
- μ_k, σ_k: Parameters that define the center and spread of the growth mapping.
- h_k: A parameter that adjusts the weight within the kernel.

2.3 Flow Lenia

Flow Lenia [4] is an extension of Lenia that incorporates a continuous velocity field into the evolution of cellular states. Unlike the original Lenia, where updates rely solely on static kernels and growth mappings, Flow Lenia introduces an advection or "flow" term that moves the density of cells according to a local velocity vector. This velocity can be derived from various sources, such as pre-defined flow equations or coupling with external forces (e.g., fluid dynamics).

In practical terms, Flow Lenia adds the following modifications:

- A velocity field $\mathbf{P}_v$ that updates the position or phase of each cell before applying the traditional Lenia kernel-based update.
- An advection-diffusion-like process, where the cell states may be transported or "flowed" across space, creating more dynamic and shifting patterns than standard Lenia.

By incorporating flow, the model is able to simulate phenomena such as traveling waves, rotating spirals, or vortex-like structures more naturally. In the context of Boid-Lenia integration, Flow Lenia could allow the Lenia field itself to move or warp in response to the motion of Boid agents, providing an additional feedback loop between the swarm and the underlying field.

2.4 Particle Lenia

Particle Lenia [5] is a derivative model that applies the principles of the original Lenia to a particle system. By replacing Lenia's grid-based interactions with dynamic particle-based interactions, this approach enables mass-conservation.

$$\frac{d\mathbf{p_i}}{dt} = -\nabla \mathbf{E}(\mathbf{p_i}) = \left[\frac{\partial \mathbf{E}(\mathbf{p_i})}{\partial p_i}\right]^t$$

Here, $\mathbf{p}_i$ represents the position of a particle in 2D space, and $\mathbf{E}$ represents the energy of the system $\mathbf{E} = \mathbf{R} - \mathbf{G}$ where $\mathbf{R}$ and $\mathbf{G}$ can be written, respectively, as follows.

$$R^t(x) = \frac{c_{\text{rep}}}{2} \sum_{i:p \neq x} \max(1 - \|x - p_i\|, 0)^2$$

where c_{rep} is a parameter that controls the strength of repulsion between particles. R^t is called repulsion potential field, where the particle distance is less than 1, the distance is multiplied by c_{rep}, and the contributions from all particle pairs are summed. Namely, particles with a distance of 1 or more are not repulsive.

$$\begin{aligned}\mathbf{G}^t(\mathbf{x}) &= G(\mathbf{U}^t(\mathbf{x})) \\ &= G\Big(\sum \mathbf{K}(\mathbf{x} - \mathbf{p}_i^t)\Big) \\ &= G\Big(\sum K(\|\mathbf{x} - \mathbf{p}_i^t\|)\Big)\end{aligned}$$

where $G(u) = \exp\Big(-\frac{(u-\mu_G)^2}{\sigma_G^2}\Big)$ and $K(r) = w_K * \exp\Big(-\frac{(r-\mu_K)^2}{\sigma_K^2}\Big)$. K corresponds to the adjacent cell rule in the game of life and determines which adjacent particle is used for the next update of its own state; G corresponds to the growth function in the Game of Life and determines its next state. μ_* and σ_* in K and G weight the states of other particles for use in updating its own state. w_K is chosen so that the integral of K over the whole space equals one. Particle will minimize repulsion energy R, and maximize growth energy G. They are modeled without momentum, behaving similarly to microscopic organisms.

Chapter 3

Integration of Boid and Lenia

Both the Boid model and Lenia exhibit life-like qualities, albeit in different ways. The Boid model emulates flocking behaviors seen in nature, where individual agents interact through cohesion, alignment, and separation to produce collective motion reminiscent of birds or fish. Lenia, on the other hand, generates geometric patterns in a continuous cellular automaton system, driven by kernel-based local influences. By integrating these two models, we aim to merge the dynamic, interactive nature of Boid with the emergent, self-organizing patterns of Lenia, creating a system capable of expressing even more complex and diverse behaviors.

3.1 Boid-Lenia Dynamics

Our proposed approach, inspired by classical models in physics and chemistry, shares similarities with Particle Lenia. A notable example is the Lennard-Jones potential [6], characterized by a system that combines attractive and repulsive potential fields.

Lenia's ring-shaped kernel introduces new behavioral rules to the Boid model. Specifically, it modifies interactions between individuals by weighting them through the ring kernel, resulting in the following behavioral changes:

- **Ring-shaped influence range**: The range of influence is no longer a simple fixed distance but takes on a ring shape, providing different effects at the center and the periphery.
- **Growth and dissipation**: By applying Lenia's growth rate to Boids, it becomes possible to simulate behaviors where the population or density dynamically changes under specific conditions.
- **Utilization of peripheral information**: Boids can leverage Lenia's properties to broaden their field of view while selectively responding to critical sources of influence.

This integration creates a framework where swarm dynamics and life-like cellular behavior coexist, leading to emergent properties not achievable with either model in isolation.

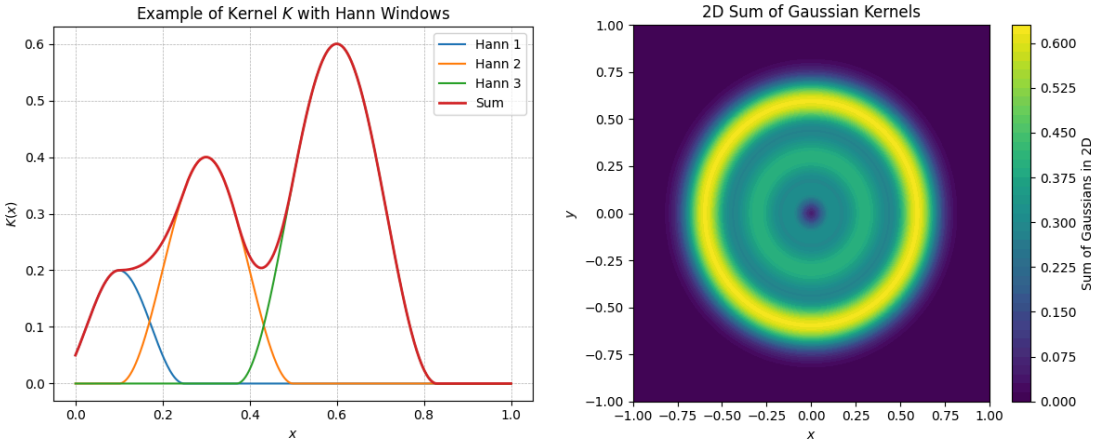
3.2 Definition of Kernels

The shape of the kernel follows the approach in [7], representing a sum of multiple window function. This design makes the parameters differentiable, which is expected to facilitate learning.

$$K(x) = \sum_{i}^{k} w_{\text{hann,i}}$$

$$w_{\text{hann}}(x) := \begin{cases} \frac{1}{2} b_i \left(1 + \cos\left(\pi \frac{x - a_i}{w_i}\right)\right) & \text{if } \left|\frac{x - a_i}{w_i}\right| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

To stabilize GPU computation and smoothly fade out the Lenia field, we apply a Hann window function, where $r_{\max}$ is a cutoff radius. This ensures that growth contributions vanish near the boundary of the kernel, reducing abrupt transitions.



a) Kernel value b) Kernel map on 2D space

Figure 1: Examples of kernels

3.2.1 Attenuating Growth

In the Boid model [1], the forces of Cohesion and Alignment are divided by the number of individuals in the vicinity. Following this principle, our model defines the force field as follows:

$$\mathbf{v}^{t+\Delta t} = \left[\mathbf{v}^t + \Delta t \mathbf{G}(\mathbf{x}^t)\right]_{\text{scaled}}$$

Here, the scaling process $[\cdot]_{\text{scaled}}$ is defined as:

$$\mathbf{v}_{\text{scaled}} = \mathbf{v} \cdot \tanh\left(\frac{\|\mathbf{v}\|}{v_{\max}}\right) * \gamma$$

where $v_{\max}$ is the maximum velocity limit, and γ is the scaling coefficient.

In the integrated Boid-Lenia model, a velocity limit is introduced to prevent individuals from accelerating excessively. The hyperbolic tangent (tanh) function is employed to smoothly limit the velocity, avoiding abrupt changes. Additionally, γ introduces a damping effect similar to air resistance, enhancing the realism of the behavior.

At each time step, the gradient $G(\mathbf{p})$ at position $\mathbf{p}$ is calculated:

$$\mathbf{G}_{\text{sep}}(\mathbf{p}) = \sum_{i \in P} \frac{\mathbf{x}_i - \mathbf{p}}{\|\mathbf{x}_i - \mathbf{p}\|} (r_{\text{kernel}} - \|\mathbf{x}_i - \mathbf{p}\|)^2$$

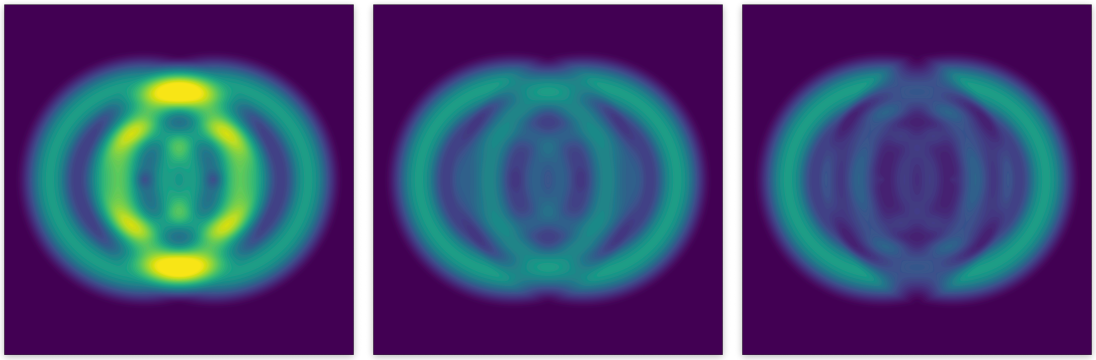
$$\mathbf{G}_{\text{align}}(\mathbf{p}) = \sum_{i \in P} \mathbf{v}_i \frac{\left(\sum_i \mathbf{K}(\mathbf{x}_i - \mathbf{p})\right)^{d_{\text{align}}}}{\sum_i \mathbf{K}(\mathbf{x}_i - \mathbf{p})^{d_{\text{align}}}}$$

$$\mathbf{G}_{\text{cohe}}(\mathbf{p}) = \sum_{i \in P} (\mathbf{x} - \mathbf{p}) \frac{\left(\sum_i \mathbf{K}(\mathbf{x} - \mathbf{p}_i)\right)^{d_{\text{cohe}}}}{\sum_i \mathbf{K}(\mathbf{x} - \mathbf{p}_i)^{d_{\text{cohe}}}}$$

$$\mathbf{G}(\mathbf{p}) = c_{\text{sep}} \mathbf{G}_{\text{sep}}(\mathbf{p}) + c_{\text{align}} \mathbf{G}_{\text{align}}(\mathbf{p}) + c_{\text{cohe}} \mathbf{G}_{\text{cohe}}(\mathbf{p})$$

By adjusting the attenuation coefficients d_{align} and d_{cohe} based on density, we regulate the changes in growth rates influenced by density.

As shown in Figure 2, setting d_{cohe} to a value greater than 1 demonstrates how the force fields counteract each other.



a) $d_{\text{cohe}} = 1.0$ b) $d_{\text{cohe}} = 2.0$ c) $d_{\text{cohe}} = 3.0$

Figure 2: Kernel value heatmap for different density attenuation parameters

3.3 Multi-channel

As shown in Algorithm 1, we employ a multi-channel approach to allow particles (Boids) of different types or "channels" to interact distinctly. For example, a swarm of predator-type agents and a swarm of prey-type agents could be treated as separate channels, each with its own weighting factors for separation, alignment, and cohesion. The key points of this algorithm are:

- Particles are assigned to grid cells for efficient neighborhood look-up, reducing the computational cost from $O(N^2)$ to approximately $O(N)$ in sparse environments.
- For each particle, we compute the gradient by summing the contributions of particles in adjacent cells, weighted by the multi-channel matrix `Weights[i.channel][j.channel]`.
- After updating velocities, we clamp them to a maximum speed (`MAX_SPEED`) to prevent numerical instability and to better mimic real swarm dynamics.
- Periodic boundary conditions (mod 1.0) ensure particles re-enter on the opposite side when they exit the simulation space.

This channeled approach provides flexibility in modeling interactions between heterogeneous agents, which can be extended to incorporate Lenia-like kernels for additional complexity.

Algorithm 1: Multi-channel Boid Algorithm

```
Assign particles to grid cells:
for each boid i in Particles:
    Compute grid index g based on p_i
    Atomically increment g.count
    Add i to g.particleIndices
end
Compute gradients and update boids:
for each boid i in Particles:
    Initialize gradient ⟵ (0, 0)
    Identify current grid cell g and neighboring cells N(g)
    for each cell c in N(g):
        for each boid j in c.particleIndices:
            if i ≠ j then
                Compute distance d ⟵ ||p_i - p_j||
                Compute separation force S based on inverse-square law
                Compute alignment force A using kernel functions
                Compute cohesion force C using kernel functions
                gradient ⟵ gradient + (S + A + C) × Weights[i.channel][j.channel]
            end
        end
    v_i ⟵ v_i + gradient
    Limit v_i to v_{max}
    Apply damping: v_i ⟵ v_i × γ
    p_i ⟵ (p_i + v_i × Δt) mod 1.0
end
```

3.4 Implementation Details

Our simulation framework is implemented in C++ with CUDA for GPU acceleration. Each time step consists of two major kernels:

1. **Grid Building Kernel**: Particles are assigned to uniform grid cells based on their (x, y) positions. This step is essential to limit the neighbor search to a few adjacent cells rather than the entire particle set.
2. **Force Computation & Integration Kernel**: For each particle, we compute separation, alignment, cohesion forces (Boid model) and Lenia-based kernel influences in parallel. We then update the position and velocity of each particle by integrating these forces.

The grid cell size is chosen to match the maximum interaction radius among particles, which is the radius of the Lenia kernel r_k. We also leverage shared memory to cache the positions of particles in the same cell, minimizing global memory transactions. As a result, the simulation scales to thousands or tens of thousands of particles in real time on a standard GPU. These optimizations enable us to perform parameter sweeps and random searches more efficiently.

Chapter 4

Experiments

4.1 Setup and Methodology

The initial state, including the number of Boids and Lenia particles, their positions, and initial velocities, was randomly generated. The two-dimensional space in which particles move is toroidal, meaning the position of a particle $\mathbf{x} = (x, y)$ follows periodic boundary conditions in both dimensions, where $x, y \in [0, 1)$. This configuration allows individuals to move freely without concern for boundary conditions, making it suitable for observing natural flocking behavior and particle evolution.

This setup prevents particles from moving infinitely far apart, ensuring that their interactions remain observable throughout the simulation.

4.2 Visualization

The visualization of the Boid-Lenia system is as follows:

- In the Boid model, velocity vectors are critical for identifying the state of individual agents. Therefore, we visualized the velocity vectors using the HSV color space, which effectively highlights the characteristics of each Boid.

As part of the visualization, we present:

1. A diagram where particles are colored in red.
2. A corresponding diagram illustrating the same state, represented in the HSV color space for velocity visualization.

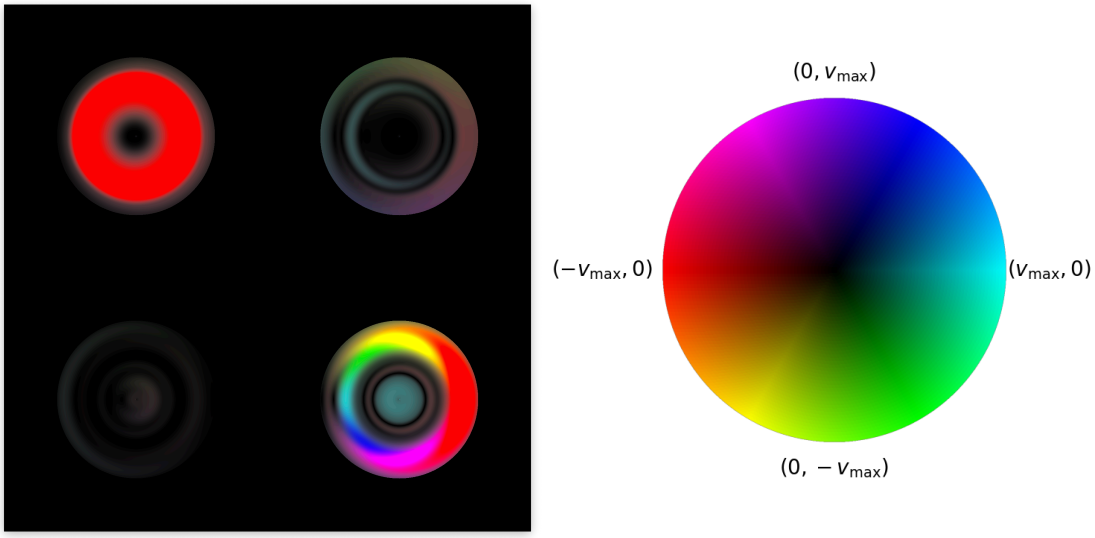
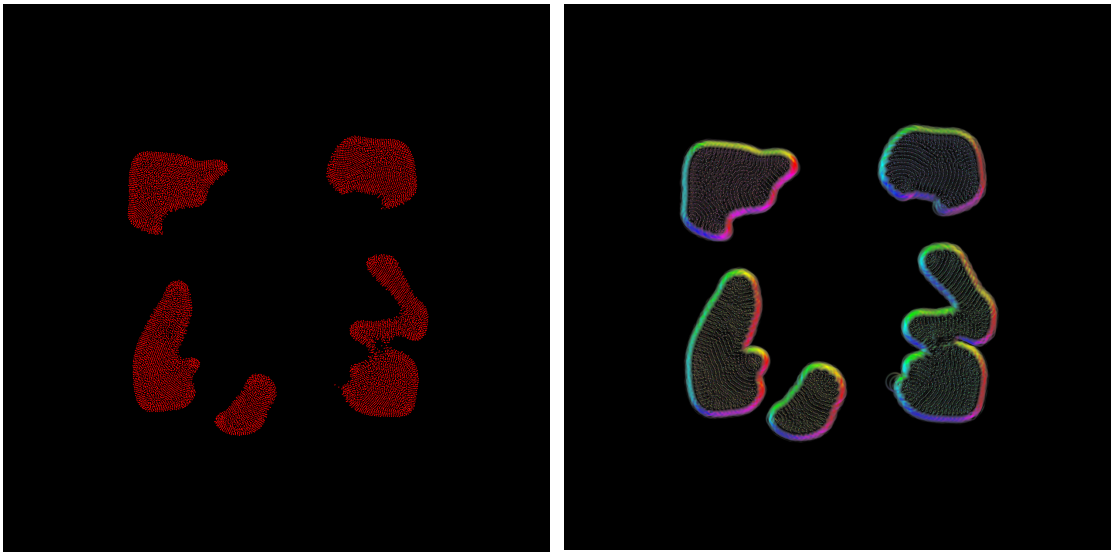


Figure 3: 2x2 grid of Lenia kernels defining the interactions between two channels

- Top-left: Self-interaction of channel 0 (channel 0 ⟶ channel 0).
- Top-right: Influence of channel 0 on channel 1 (channel 0 ⟶ channel 1).
- Bottom-left: Influence of channel 1 on channel 0 (channel 1 ⟶ channel 0).
- Bottom-right: Self-interaction of channel 1 (channel 1 ⟶ channel 1).

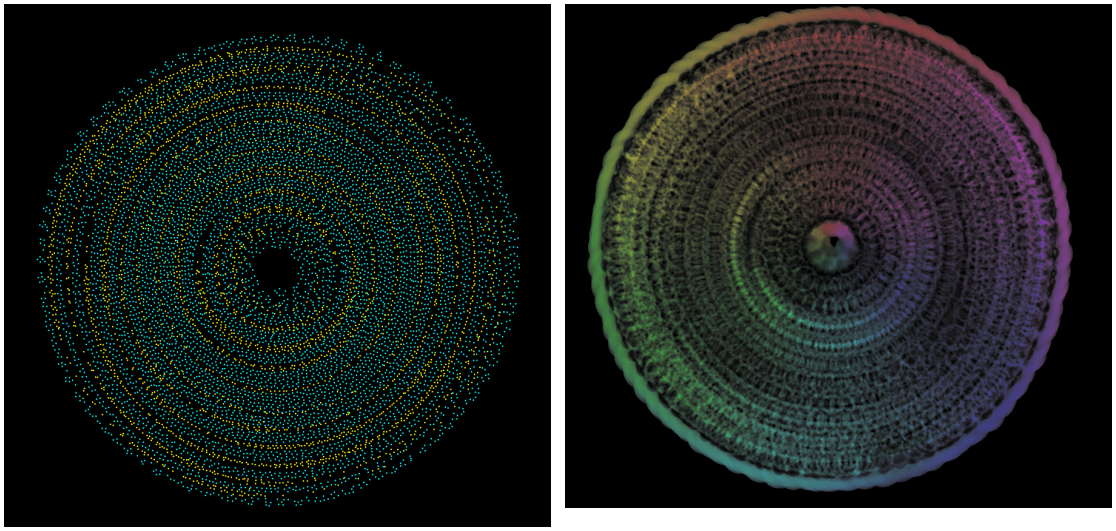
Figure 3 illustrates the force field generated around a particle moving at maximum velocity in the rightward direction. The direction of the force field is represented in HSV color space, following the mapping shown in the adjacent legend. Blue represents a forward-directed force field, while red indicates a backward-directed force field.

From this point onward, boids will be represented as red particles in single-channel environments, and as cyan and yellow particles in two-channel environments.



a) Channel 0 described by red particle b) with hsv

Figure 4: “1-channel boids”



a) particle b) gradient

Figure 5: 2-channel boids

Chapter 5

Evaluation Methods

5.1 Evaluation Criteria

To evaluate the integrated Boid-Lenia model, the following criteria were established:

1. **Overall system stability**
   The stability of the system is evaluated to ensure that the interactions between Boid and Lenia do not lead to excessive destabilization, maintaining a dynamic equilibrium.

- **Particle distribution**: Although the initial distribution of particles is random, clustering or uniformity emerges over time, influenced by Boid's cohesion and Lenia's growth rates.
- **Toroidal space characteristics**: When particles cross the toroidal boundaries, they seamlessly reappear on the opposite side. The periodic boundary conditions are accounted for in distance calculations between particles, ensuring accurate interactions.

5.2 Calculation of Boid Center

To understand the overall movement of the Boid swarm, the center-of-mass velocity is calculated. This metric indicates how the swarm moves through space and serves as a key indicator of how interactions with Lenia influence Boid dynamics.

In toroidal space, particle positions are restricted to the range $[0, 1) \times [0, 1)$. Considering these periodic boundary conditions, the calculation of the center-of-mass and dispersion employs the Circular mean method.

Given a particle position $\mathbf{p}_i = (x_i, y_i)$, the center in toroidal space is computed as follows: In toroidal space, each coordinate component x_i and y_i is converted to an angular position $\theta_i = 2\pi x_i$ on a circle.

1. Treating particle positions as angles on a circle, the mean angle is calculated using the sine and cosine averages for each component:

$$x_{\text{center}} = \text{atan2}\left(\frac{1}{N}\sum_{i=1}^{N}\sin(2\pi x_i), \frac{1}{N}\sum_{i=1}^{N}\cos(2\pi x_i)\right)$$

y_{center} is computed in a similar manner to obtain $p_{\text{center}} = (x_{\text{center}}, y_{\text{center}})$.

5.3 Variation of Particles in Torus

The variance of particles in toroidal space is computed by adjusting the distances from the center using periodic boundary conditions. The distance of each particle from the center is calculated while accounting for periodicity:
$\Delta\mathbf{p}_i = \mathbf{p}_i - \mathbf{p}_{\text{center}} - \left\lfloor \mathbf{p}_i - \mathbf{p}_{\text{center}} + \frac{1}{2} \right\rfloor$

Here, $\Delta \mathbf{p}_i$ represents the distance of particle i from the center. The variance is defined as: $\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} \|\Delta \mathbf{p}_i\|^2$

5.3.1 Probability Density Function (KDE)

The probability density function of a Gaussian distribution is defined as follows:

$$f(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

Here, σ is the standard deviation, and $2\pi\sigma^2$ is the normalization factor for the distribution in two-dimensional space. This ensures that the integral of the distribution equals 1.

In this study, instead of using a Gaussian function as the target function, we employed a custom function $G(x_{\text{dist}})$ based on the Smoothstep function. The Smoothstep function, $\text{Smoothstep}(x) = 3x^2 - 2x^3$, has the characteristic of smoothly interpolating input values within the range of 0 to 1.

$$G(x_{\text{dist}}) := \begin{cases} 1 & \text{if } (0 \leq x_{\text{dist}} \leq R_{\text{inner}}) \\ -3\left(\frac{r - R_{\text{inner}}}{R_{\text{outer}} - R_{\text{inner}}}\right)^2 + 2\left(\frac{r - R_{\text{inner}}}{R_{\text{outer}} - R_{\text{inner}}}\right)^3 + 1 & \text{if } (R_{\text{inner}} < x_{\text{dist}} \leq R_{\text{outer}}) \\ 0 & \text{if } (R_{\text{outer}} < x_{\text{dist}}) \end{cases}$$

The reason for adopting this shape for the target function instead of a Gaussian function is to prevent particles from becoming overly concentrated at the center. Additionally, since the size of the kernel is fixed, the formation of groups is expected to exhibit periodicity. To maintain this periodicity, it was considered necessary to have density remain within a certain interval.

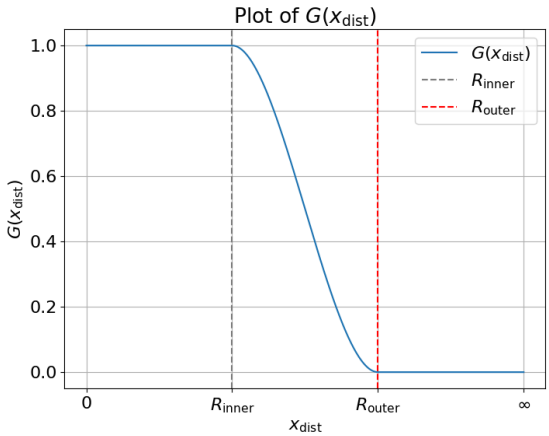
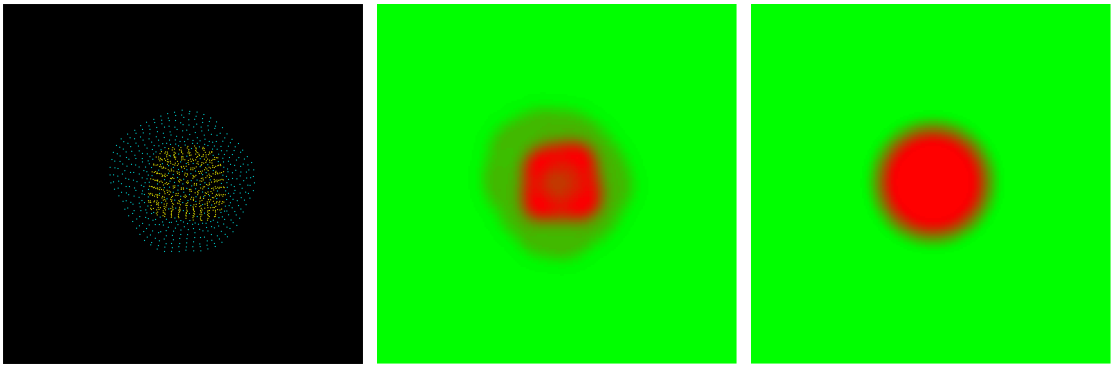


Figure 6: Target Distribution

Here, we propose two approaches to calculate the variance of Boids.

5.3.2 2-dimensional KDE

We evaluate whether the heatmap of particles in a two-dimensional space corresponds to the expected distribution. Specifically, when plotting the previously defined distribution, it appears as follows:



a) particle position b) KDE density c) target density

Figure 7: Density Heatmap

5.3.3 1-dimensional KDE

Next, we explore an approach that uses only the distance as the basis for evaluating the distribution. Unlike the previous method, this approach is not restricted to a specific shape. Instead, it examines the distribution of distances from the center.

To ensure that this distribution closely resembles the two-dimensional spatial distribution, we use the target density distribution obtained by circularly integrating the previously defined target function. The distribution resulting from integrating Equation 1 is expressed as follows:

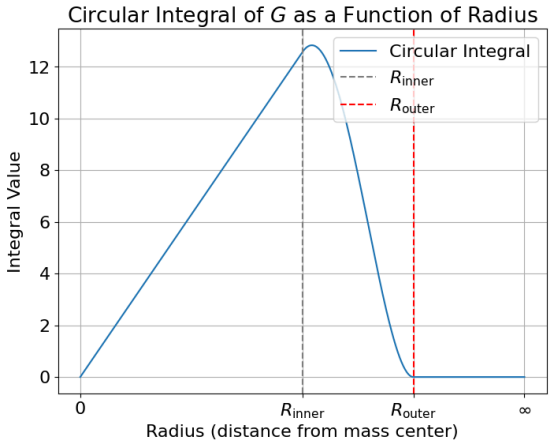


Figure 8: Density Heatmap

5.3.4 Convolution

To estimate the particle distribution using KDE, we convolve the target distribution with the same Gaussian function. This approach ensures that the estimated distribution aligns with the target distribution while smoothing out noise or irregularities.

The convolution operation is defined as follows:

$$F_{\text{Blurred}(x_0)} = (f * g)(x_0) = \int_{\infty}^{\infty} (f(t) * g(x_0 - t))\,\mathrm{d}t$$

Here:

- $f(t)$ represents the target distribution.
- $g(x)$ is the Gaussian function used for convolution.
- x_0 denotes the evaluation point for the resulting blurred distribution.

By performing this convolution, we obtain a smoothed version of the target distribution, which serves as a reference for comparing the particle distribution estimated via KDE. This step is essential for ensuring that the analysis focuses on meaningful features rather than noise.

5.4 Loss Calculations

To compare the boids distribution with the target distribution, we employed the Jensen-Shannon divergence (JSD) [8], which is a symmetric variant of the Kullback-Leibler (KL) divergence. The KL divergence, commonly used to measure differences between probability distributions, has a critical limitation: it becomes undefined when comparing distributions with zero density in certain regions. Since the boids' positions may result in highly skewed distributions with near-zero densities, the KL divergence can lead to instability or unreliable results. In this study, the JSD is used to quantify how closely the simulated boids distribution aligns with a predefined target distribution. A lower JSD value indicates higher similarity, allowing us to evaluate the effectiveness of the parameter settings in guiding the boids toward the desired spatial patterns.

Chapter 6

Results and Discussion

The radius of the kernel was set to 0.05 in $[0, 1) \times [0, 1)$ two-dimensional surface, and the number of Boids was set to 1000.

6.1 Random Search

Random search is a parameter optimization method where parameters are varied randomly within a predefined range. The goal is to find parameter values that minimize the loss function effectively.

The parameter space for this search includes:

KernelParam	Kernel bumps: Alignment	Kernel bumps: Cohesion
$c_{\text{sep}}, c_{\text{align}}, c_{\text{cohe}} \in [0, 1]$	$a \in [0, 1]$	$a \in [0, 1]$
$d_{\text{align}}, d_{\text{cohe}} \in [1, 3]$	$b \in [-1, 1]$	$b \in [0, 1]$
	$w \in [0.1, 1]$	$w \in [0.1, 1]$

Table 1: Parameter space

To ensure stable and coherent behavior in the flock, we carefully define parameter ranges for alignment, kernel bump size, and cohesion density. The maximum alignment strength is limited to prevent undesirable behaviors; excessively large values can cause agents to overreact to others' movements, leading to rapid oscillations that stabilize into a deadlock, where agents cease meaningful motion. For the kernel bump size, an overly narrow bump is avoided because it reduces the region of influence, which can make SLPs difficult to achieve. Finally, a minimum cohesion density greater than 1 is introduced to prevent agents from collapsing into a single overly dense cluster. While this threshold may vary with the total number of particles, our experiments showed that a value of at least 1.5 is necessary to counteract excessive aggregation. However, setting this parameter too high increases the risk of over-dispersal, potentially leading to fragmentation.

For instance, in the Boid-Lenia model, parameters such as c_{sep} and μ influence the swarm behavior and growth field interactions. By exploring values within [0, 1], the random search ensures sufficient diversity to discover configurations that balance swarm dynamics with the underlying growth patterns.

The features "Inner Energy" and "Directed Velocity" did not transition from one to the other during gradient-based learning. This observation suggests that these states are separated into distinct local optima, making it unlikely for the system to move between them under the current optimization framework.

Additionally, while the minimization of loss via KDE was achieved, its primary goal was to prevent excessive dispersion or aggregation. As such, random search was deemed sufficient for exploration in this case, since it adequately addressed the problem without requiring more complex optimization methods.

6.2 Observed Patterns

By integrating Boid and Lenia into a unified simulation model, we observed novel swarm behaviors that were not achievable with traditional Boid models. The Boid agents were influenced by Lenia's ring-shaped kernels, exhibiting tendencies to be attracted to other particles within a specific range. As a result, Lenia's characteristic growth patterns manifested in distinctive structural formations.

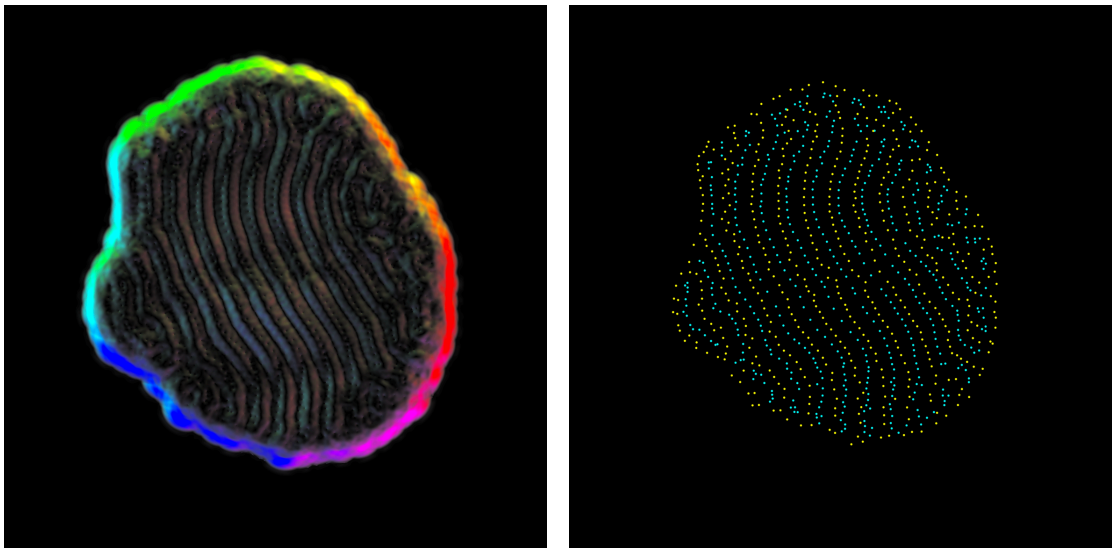
Specifically, the following features were identified:

1. In regions where Lenia's growth field was high, Boid velocities decreased, leading to stronger aggregation of the boids flock.
2. When Boids maintained movement in a specific direction, Lenia's growth patterns were dragged along, resulting in asymmetric distributions.
3. Over extended simulations, a dynamic equilibrium between Boid and Lenia behaviors was maintained under certain conditions.

These behaviors demonstrate that Boid-Lenia agents, unlike those in traditional interaction models, are dynamically modulated by Lenia's nonlinear growth processes. This integration highlights the potential of the combined model to simulate complex, emergent behaviors that bridge swarm dynamics with cellular growth mechanisms. Based on our observations, we categorized these behaviors into four distinct patterns: Flutter Pattern, Ring Pattern, Swap Pattern, and Stable Pattern.

6.2.1 Flutter Pattern

The Flutter Pattern, as shown in Figure 9, is characterized by short-period oscillations. This is the only observed pattern that forms stripe-like structures. Within a single channel, boids align in the same direction, while interactions between the two channels involve alternating phases of repulsion and attraction. These dynamics give rise to wave-like, oscillatory movements, creating a visually distinct and temporally dynamic behavior.

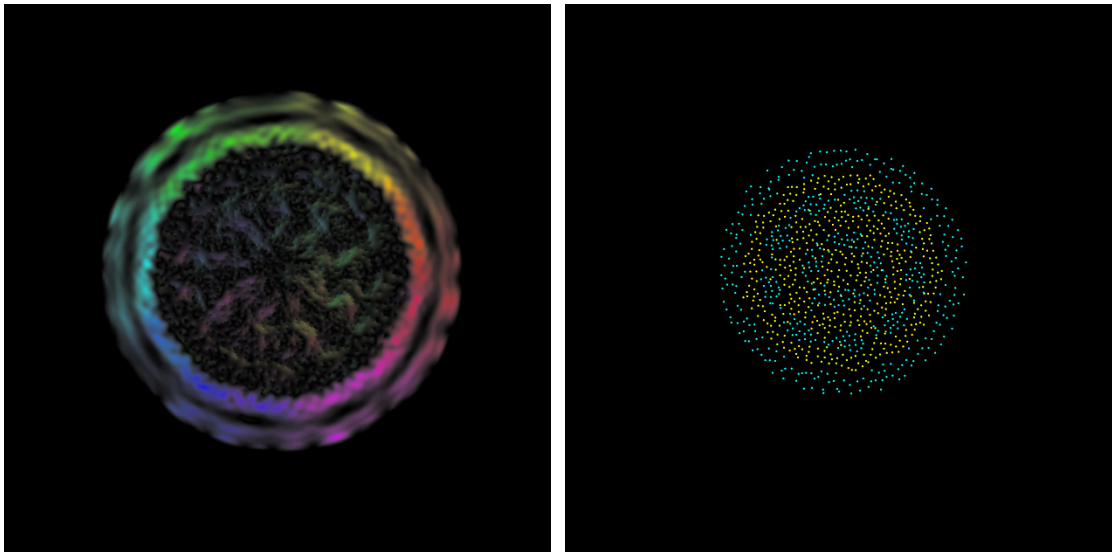


a) gradient of channel 1 b) particle

Figure 9: Flutter Pattern

6.2.2 Ring Pattern

The ring pattern is formed when boids in two channels move in opposite directions. Alignment within the same channel is positive, while alignment between different channels is negative, resulting in this behavior. This pattern exhibits continuous rotation.

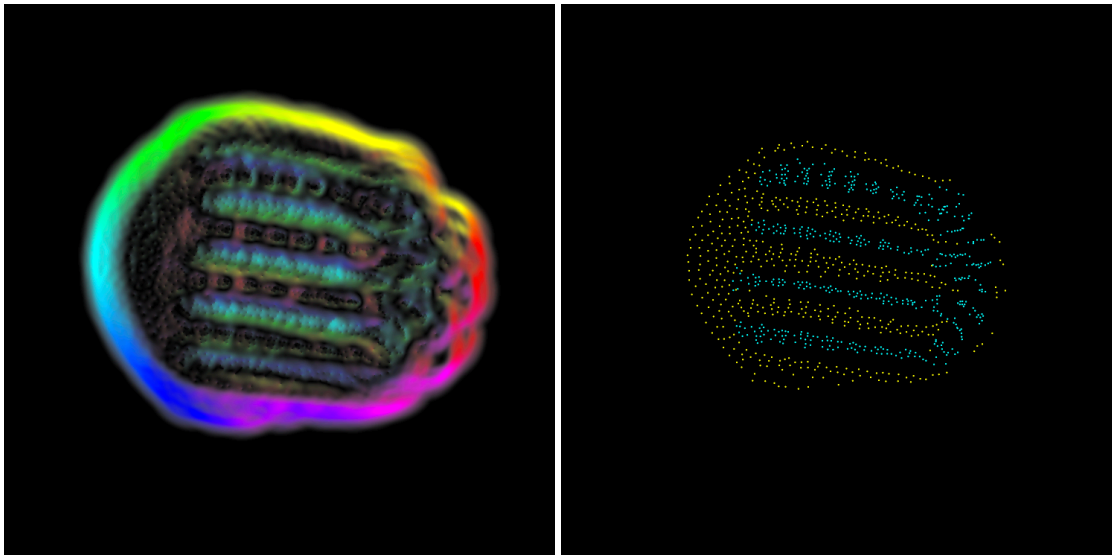


a) gradient of channel 1 b) particle

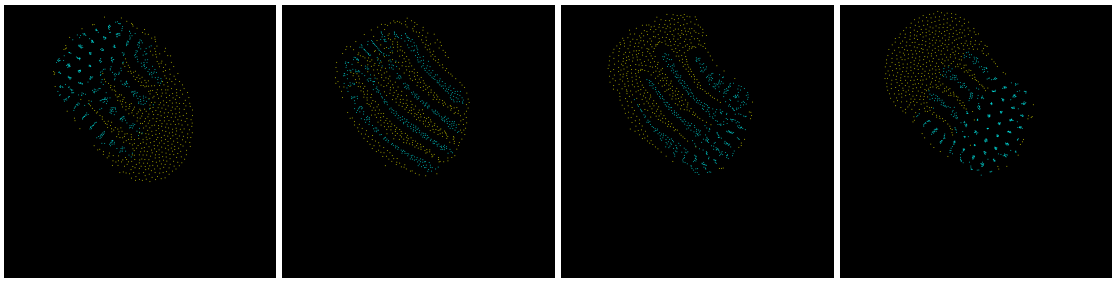
Figure 10: Ring Pattern

6.2.3 Swap Pattern

As shown in Figure 11, the boids in two channels exhibit periodic swapping movements. The interaction between the swapped channels is fundamentally similar to that of the ring pattern; however, specific dynamics result in this alternating behavior.



a) gradient of channel 1 b) particle

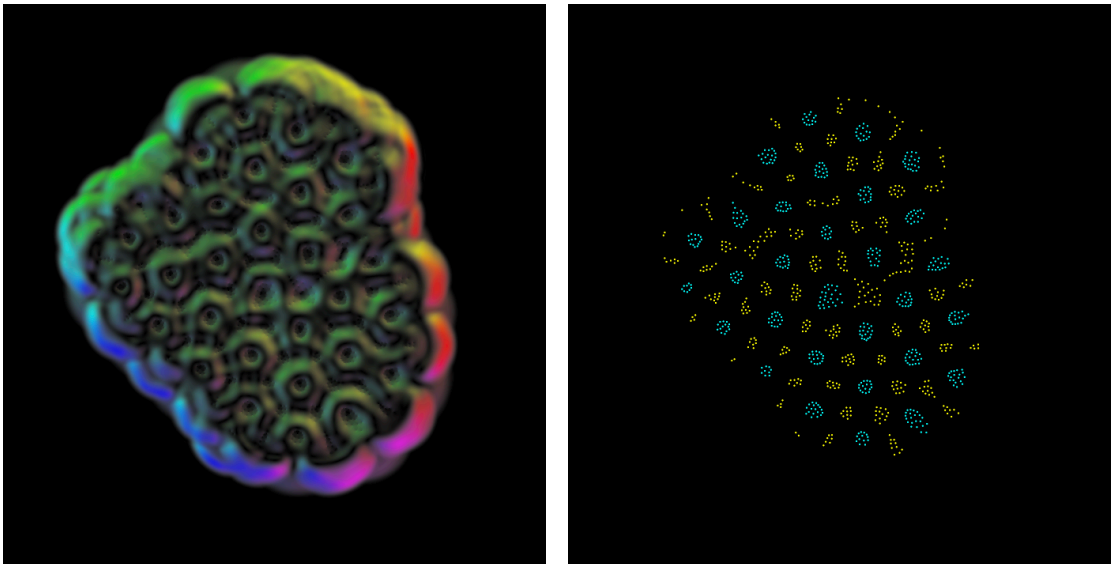


(c) (d) (e) (f)

Figure 11: Swap Pattern: The progression of timesteps is shown in the order of (c) → (d) → (e) → (f). Notably, the positions of Channel 1 and Channel 2 are reversed between (c) and (f), and this periodic swapping motion continues indefinitely.

6.2.4 Stable Pattern

The Stable Pattern is defined by minimal internal movement within the flock, even when the flock itself is in motion. Unlike the other patterns, which display continuous or periodic shifts, this pattern maintains a relatively fixed shape. However, the forms it produces exhibit greater diversity compared to other patterns. This pattern tends to arise when the negative weighting of alignment between channels is low, allowing for more cohesive, stable groupings.



a) gradient of channel 1 b) particle

Figure 12: Stable Pattern

This pattern is a stable pattern, which is a pattern that does not relatively change over time.

6.3 Movement Characteristics

Among the four observed patterns, certain Stable Patterns displayed greater displacement of the flock's center of mass, whereas Flutter, Ring, and Swap Patterns tended to remain more localized. However, not all Stable Patterns exhibited this trait; many showed only minimal overall movement. These observations point to a possible trade-off between large-scale translational motion and the internal oscillations or periodic rearrangements typical of other patterns, though further analysis is required to clarify this relationship.

6.4 Energy Analysis

We defined the total system energy as $E_{\text{flock}} = \sum_i^N \|\mathbf{G}(\mathbf{p}_i)\|$, following the approach described in Particle Lenia [5]. Also, kinetic energy is calculated as the sum of the magnitude of the velocities of all particles in the system, defined as $V_{\text{flock}} = \sum_i^N \|\mathbf{v}_i\|$) This definition was employed to evaluate the stability of the model.

The following figures illustrate the movement of the center of mass and the progression of total energy in the switcher model, respectively.

As observed in Figure 13, the total energy stabilizes significantly after certain timesteps from the start of the simulation.

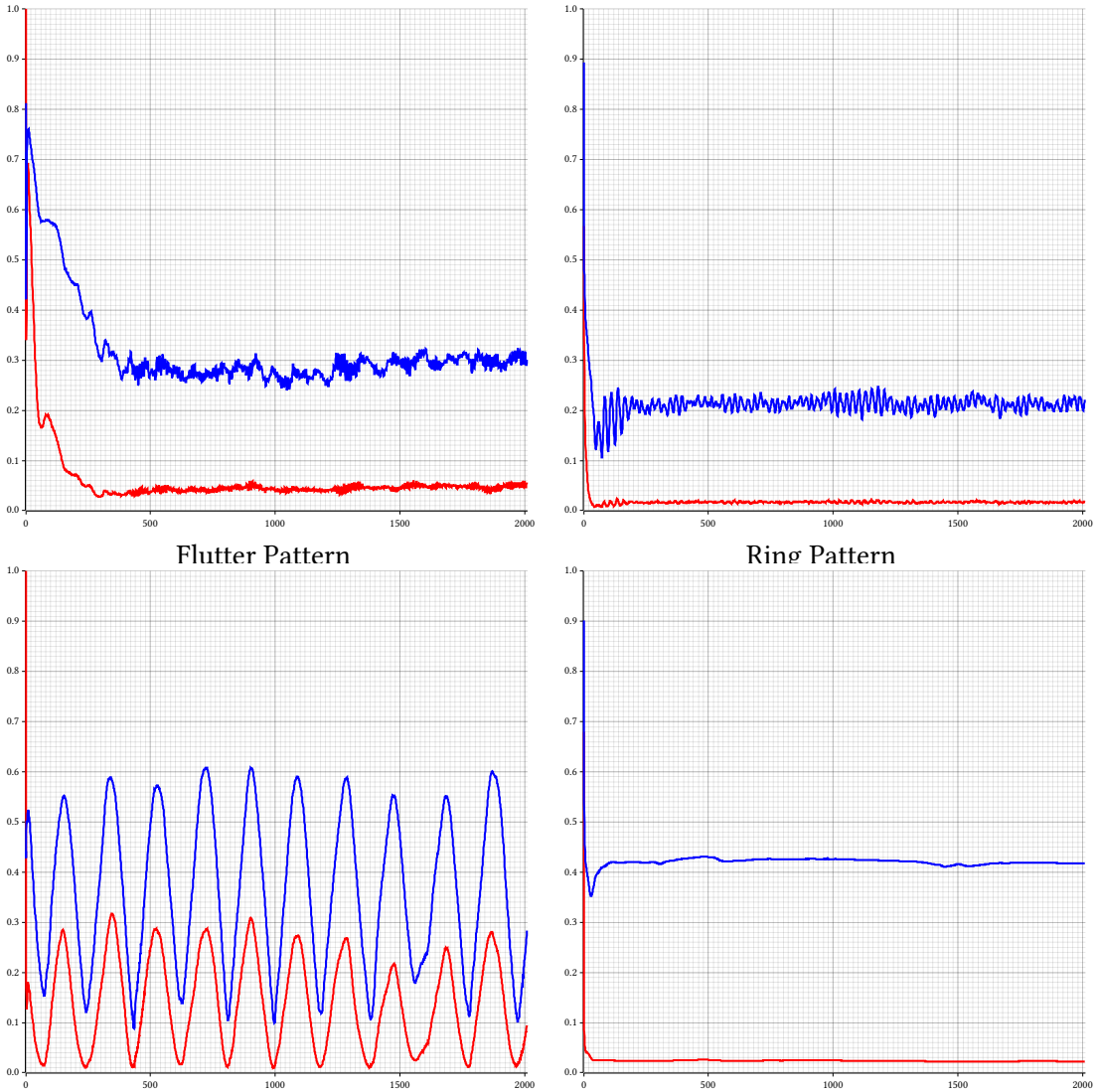
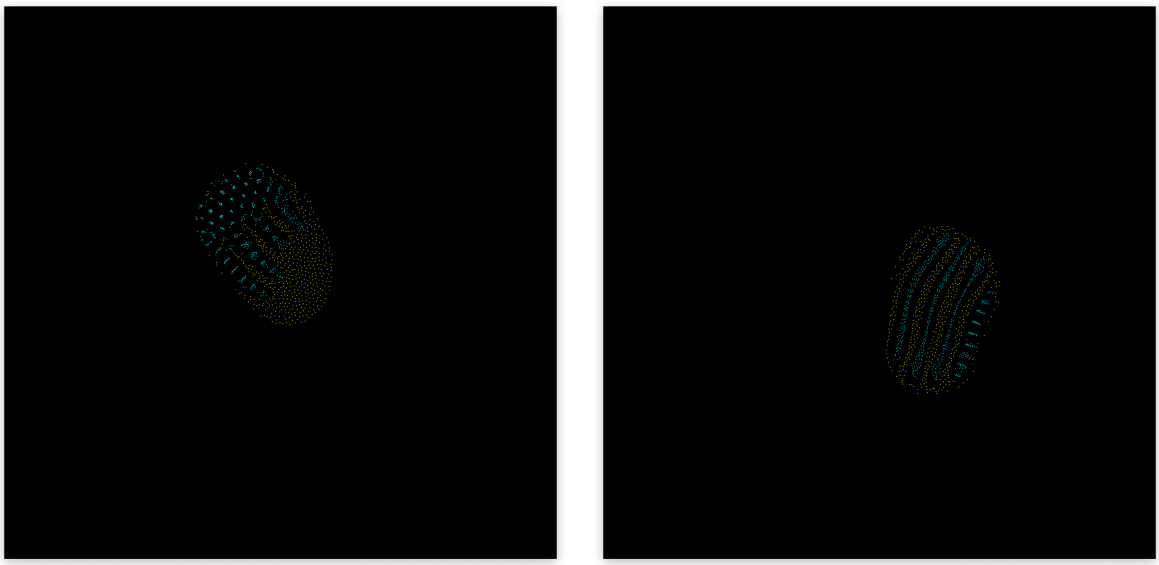


Figure 13: Energy and velocity sum over time: The red line represents the sum of gradient magnitudes E_{flock}, while the blue line corresponds to the total sum of the magnitudes of velocities V_{flock} for all boids. The x-axis represents time steps, and the y-axis indicates the magnitude of the respective energy values.

Unlike Particle Lenia, the alignment mechanism in this model allows larger velocities to create steeper gradients, resulting in behaviors that do not adhere to the law of energy conservation. Consequently, $E_{\text{flock}} + V_{\text{flock}}$ does not remain constant. However, as demonstrated in the figure, the system exhibits remarkable stability despite this deviation.

Comparing the states at timestep = 100 and timestep = 10,000, as shown in Figure 14, periodic changes within the system persist. The model exhibits the “swap” movement repeatedly, while the overall structure and energy distribution remain remarkably stable.



$t = 100$ $t = 10,000$

Figure 14: Simulation at temporally separated timesteps t

6.5 Comparison of Theoretical Predictions and Experimental Results

Theoretical predictions suggested that Lenia's ring-shaped kernels would induce stable concentric growth patterns in the behavior of Boid swarms. However, experimental results revealed that the emergence of such stable patterns was highly sensitive to parameter settings.

The following key findings were observed:

1. **Impact of Growth Rate**:
   - When the growth rate was low, Boids exerted insufficient influence on Lenia particles, resulting in monotonous Lenia patterns.
   - Conversely, when the growth rate was high, Boids became overly confined to regions of high growth in Lenia, leading to a loss of collective freedom within the swarm.
2. **Impact of Velocity Limitation**:
   - With loose velocity constraints, Lenia patterns struggled to keep up with Boid movement, leading to turbulent behaviors.
   - When velocity was appropriately controlled, stable periodic patterns driven by Boid-Lenia interactions were observed.

These findings demonstrate that the interactions between Boids and Lenia are more complex than simple intuitive predictions would suggest. The results underscore the critical role of parameter tuning in shaping the emergent behaviors of the integrated model.

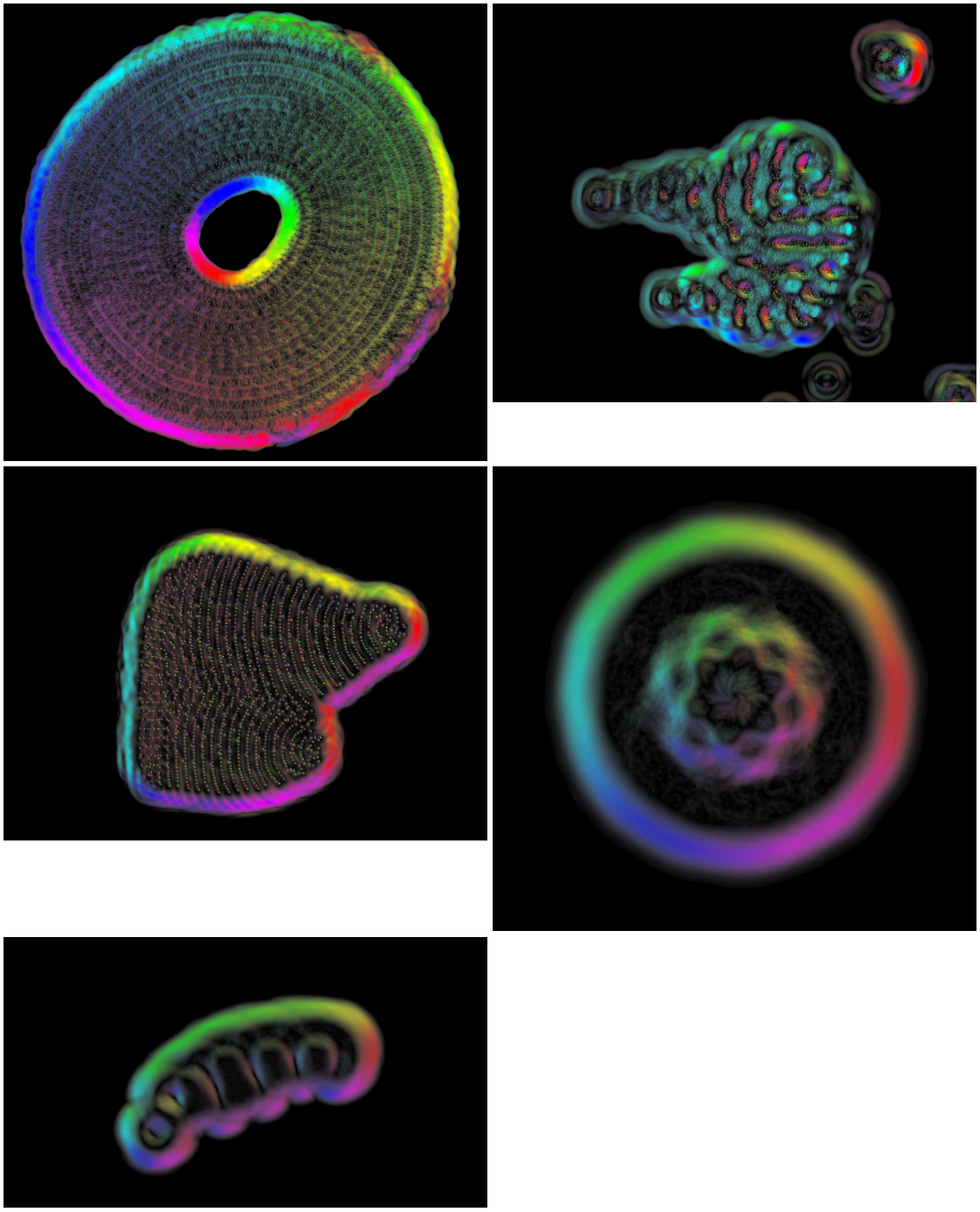


Figure 15: Collection of boid-lenia patterns

Chapter 7

Conclusion

This study demonstrated the potential for new dynamic behaviors and pattern formations through an integrated model of Boid and Lenia. In particular, the influence of Lenia's ring-shaped kernels on Boid dynamics revealed emergent behaviors not achievable with traditional models. This work proposes a novel framework that bridges swarm intelligence and self-organizing systems.

7.1 Limitations and Future Directions

1. **Exploration of a broader range of parameter settings**
   While this study focused on a limited set of parameters, investigating a more diverse range (e.g., kernel shapes, interaction ranges, and environmental noise) could uncover a wider variety of emergent behaviors. This would enhance our understanding of the adaptability and generalizability of the integrated model.
2. **Application to 3D spaces**
   Similar to the original Boid model [1], the proposed method can be extended to three-dimensional spaces. This extension could reveal dynamic behaviors closer to real-world flock phenomena and foster the discovery of novel pattern formations.
3. **Optimization of algorithms**
   To enable real-time parameter exploration, it is crucial to design algorithms that accelerate loss calculations. Such advancements would improve the practicality and scalability of the model in dynamic simulations.

- **Expanded Channel Interactions**

In the study of Lenia as a sensorimotor agency [7], simulations involve interactions with objects such as walls using additional channels. Similarly, future work on the Boid-Lenia model could focus on expanding the expected interactions between channels. While this study primarily explored patterns where two channels form a single flock, further investigations could include improving stability during collisions with fixed objects, as demonstrated in the aforementioned research. Additionally, it would be interesting to study patterns where two channels interact with each other but do not belong to the same flock.

References

[1] C. W. Reynolds, “Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH ’87,” 1987. [Online]. Available: http://dx.doi.org/10.1145/37401.37406

[2] S. Rafler, *Generalization of Conway's "Game of Life" to a continuous domain -SmoothLife*. 2011.

[3] B. W.-C. Chan, “Biology of Artificial Life,” *Complex Systems*, vol. 28, no. 3, pp. 251–286, 2019.

[4] E. Plantec, G. Hamon, M. Etcheverry, P.-Y. Oudeyer, C. Moulin-Frier, and B. W.-C. Chan, “Flow-Lenia: Towards open-ended evolution in cellular automata through mass conservation and parameter localization.” [Online]. Available: https://arxiv.org/abs/2212.07906

[5] A. Mordvintsev, E. Niklasson, and E. Randazzo, “Particle lenia and the energy-based formulation.,” Dec. 23, 2022. [Online]. Available: https://google-research.github.io/self-organising-systems/particle-lenia/

[6] J. E. Jones and S. Chapman, “On the determination of molecular fields.—I. From the variation of the viscosity of a gas with temperature,” *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 106, no. 738, pp. 441–462, 1924, doi: 10.1098/rspa.1924.0081.

[7] G. Hamon, M. Etcheverry, B. W.-C. Chan, C. Moulin-Frier, and P.-Y. Oudeyer, “Learning Sensorimotor Agency in Cellular Automata.” [Online]. Available: https://hal.inria.fr/hal-03519319

[8] J. Lin, “Divergence measures based on the Shannon entropy,” *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, 1991.