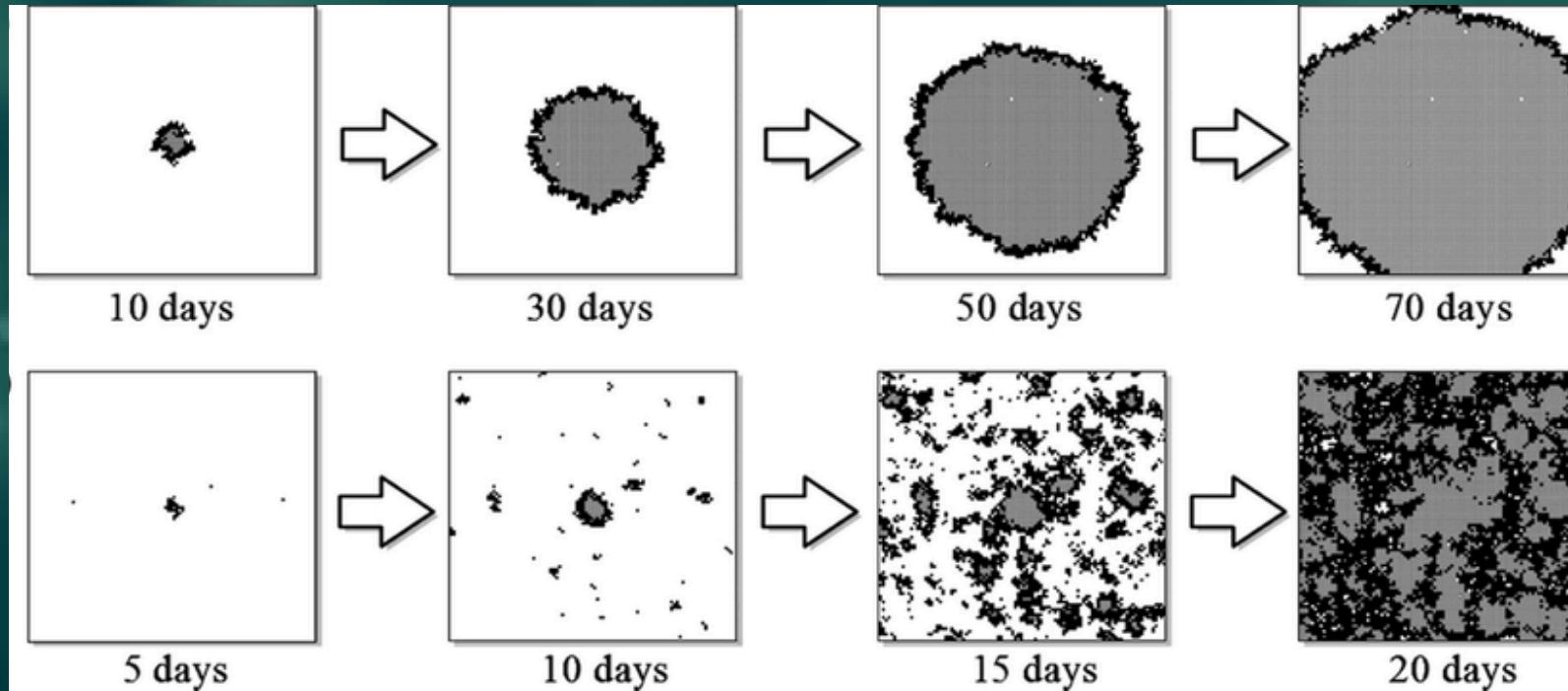


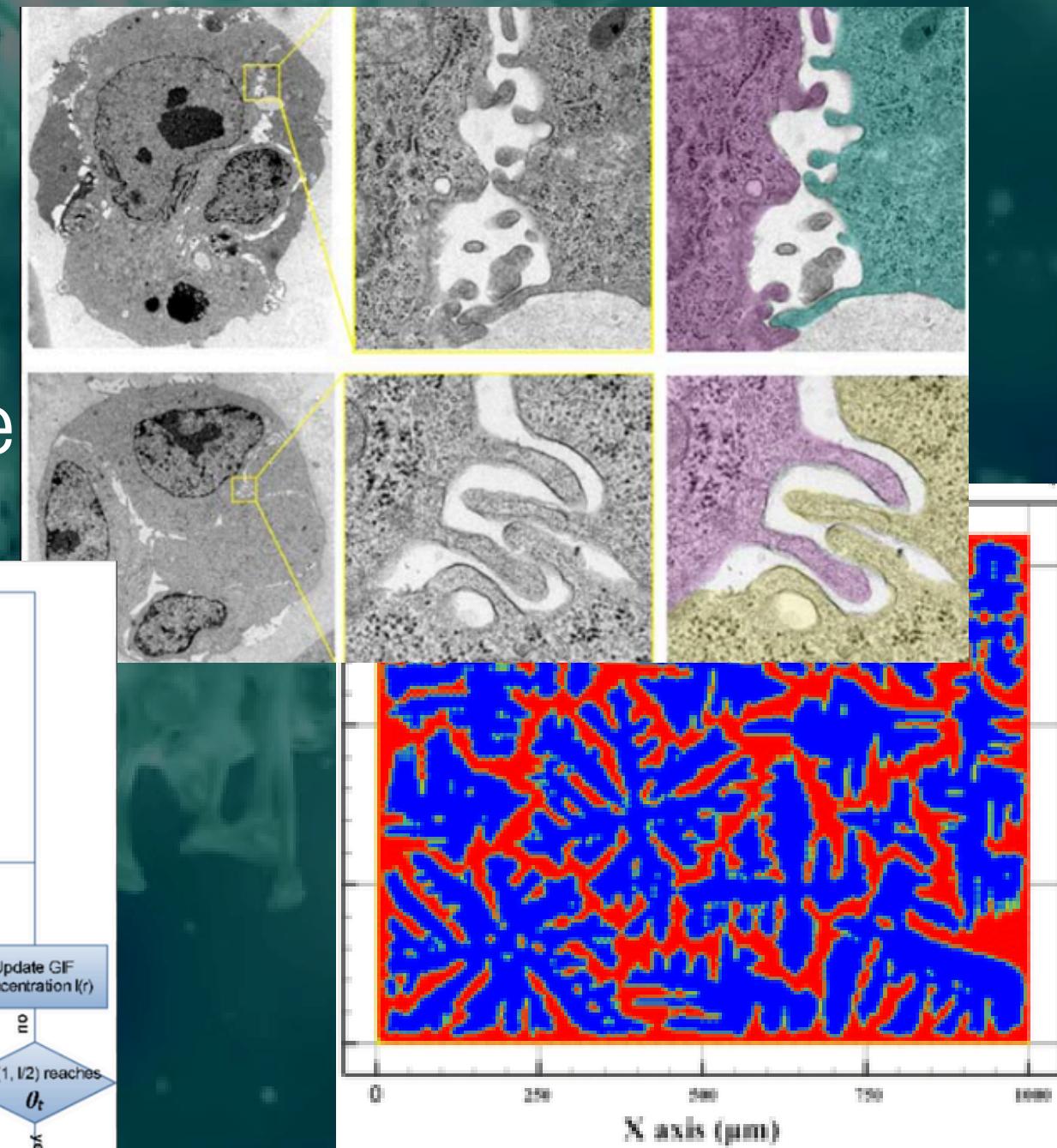
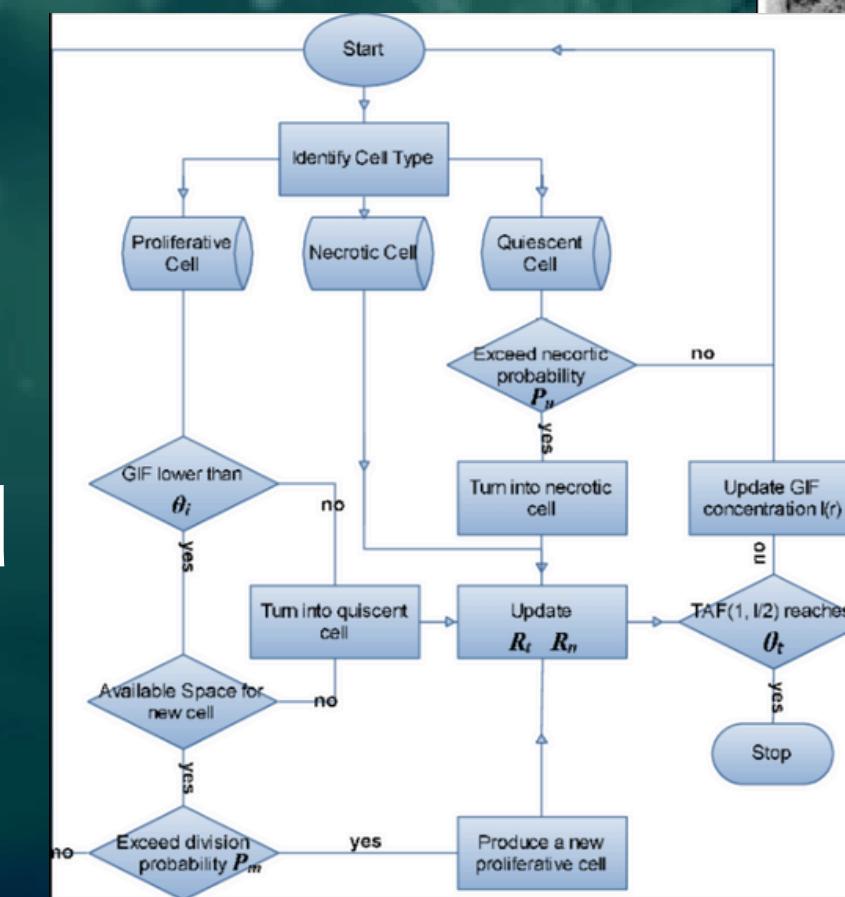
# Cellular Automata: An Algorithmic Approach to Tumor Growth



Computing the strategy of Cancer and simulating their growth and evolution.

**TEAM-4**

Sahasra Oleti  
Atharva Shetwe  
Sidhhesh Phadke



# CA-PDE Framework

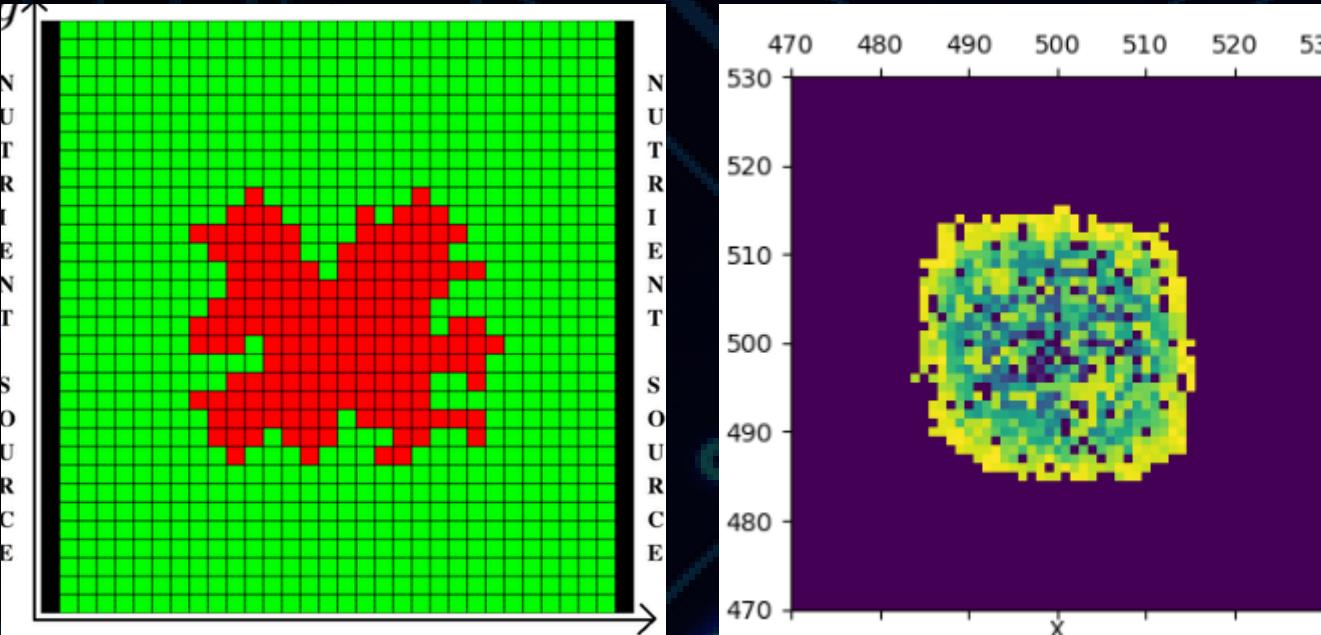


## Objective

To develop a computational framework that mimics real tumor behavior by:

- Representing cells on a grid,
- Applying biological rules for cell division, death, and spread
- Simulating nutrient availability,
- Observing how tumors grow under different conditions.

This approach helps researchers test hypotheses, analyze tumor patterns, and predict growth under controlled digital environments.



## Computational Grid

Tumor growth can be translated into a computational algorithm, where:

- Each cell becomes a grid point,
- Each update step simulates biological time,
- Biological rules become conditional operations.

In this model:

- A 2D or 3D grid represents tissue.
- Each grid cell can be:
- Empty, Tumor cell, Healthy cell, or Necrotic (dead).
- The tumor evolves based on algorithmic rules similar to how real cells behave.

## Cell States & Behavior

### Proliferating



Active tumor cells dividing when nutrients are sufficient

### Quiescent



Dormant cells in low-nutrient conditions, can reactivate

### Necrotic



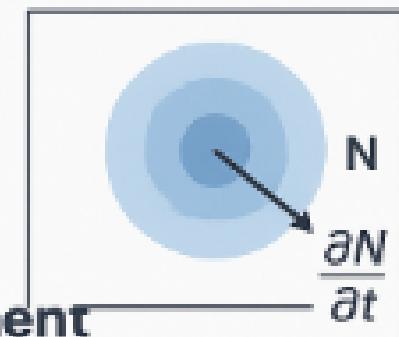
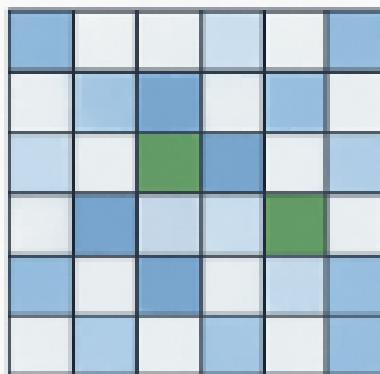
Dead cells from starvation, forming the tumor core

## CA-PDE Hybrid Model for Tumor Growth

The CA-PDE model combines discrete cell behavior (CA) with continuous nutrient diffusion (PDE) to simulate realistic tumor growth.

### Cellular Automata (CA) Component

- Represents tissue as a grid updated step-by-step.
- Each cell's next state depends on neighbors and simple biological rules.
- Typical rules: tumor cell divides if nutrient  $\geq$  growth threshold
- Captures local interactions, spatial patterns, and growth dynamics



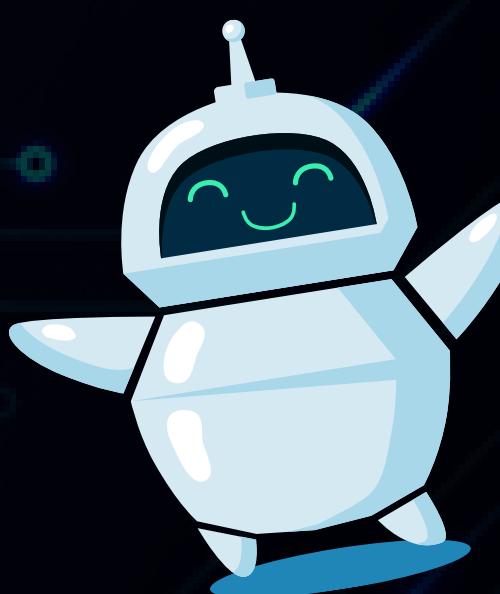
### Partial Differential Equation (PDE) Component

- Models how oxygen/nutrients diffuse through tissue
- Uses an equation like:
$$\partial N / \partial t = D \nabla^2 N - \text{consumption\_rate} \times \text{Tumor}(x,y)$$
- Calculates nutrient levels at each grid point
- Reflects diffusion, consumption, and gradients

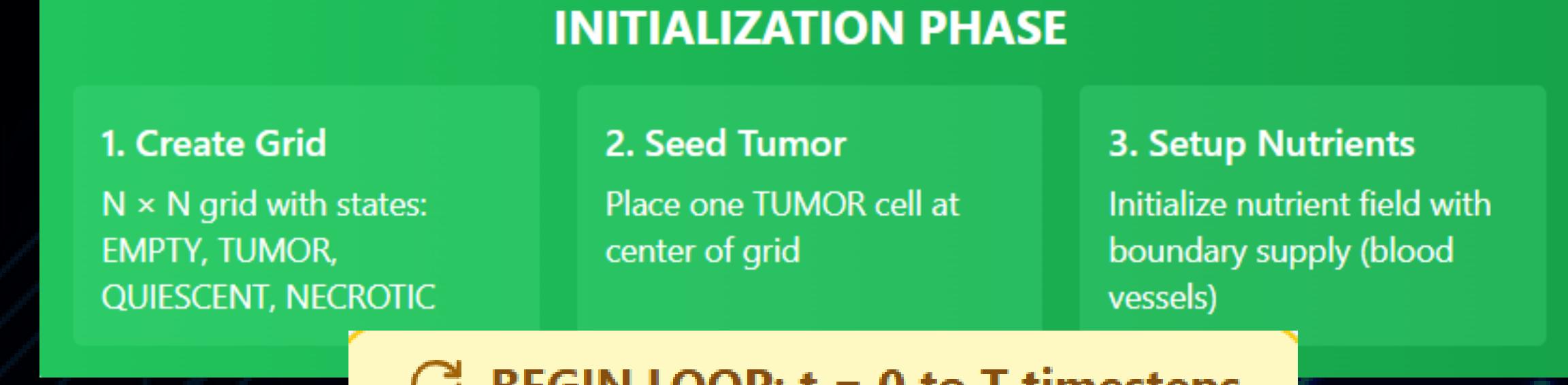
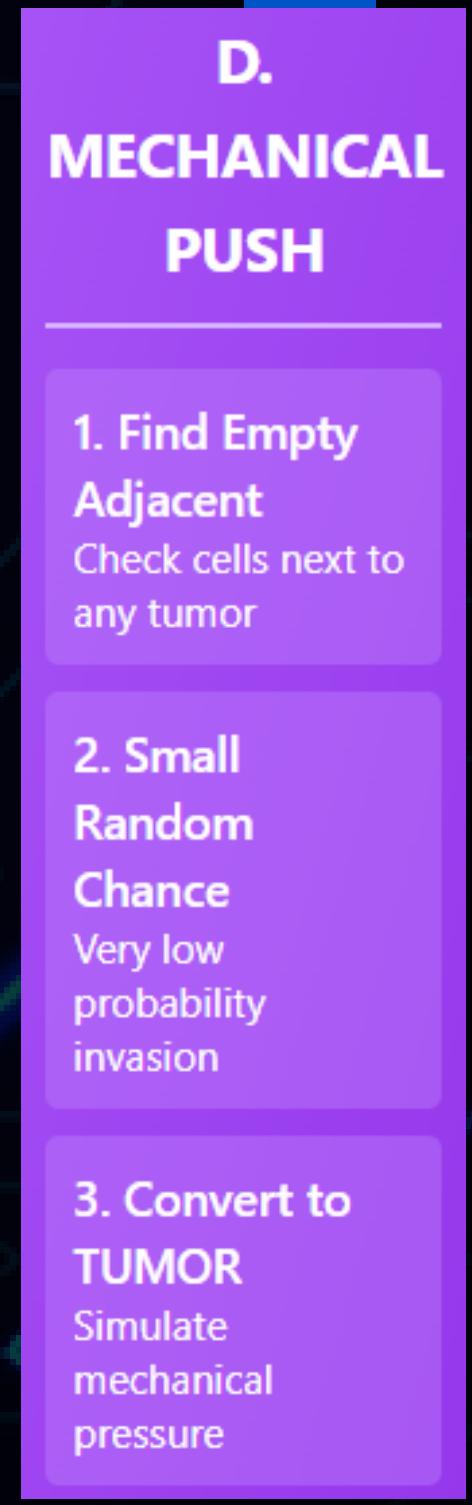
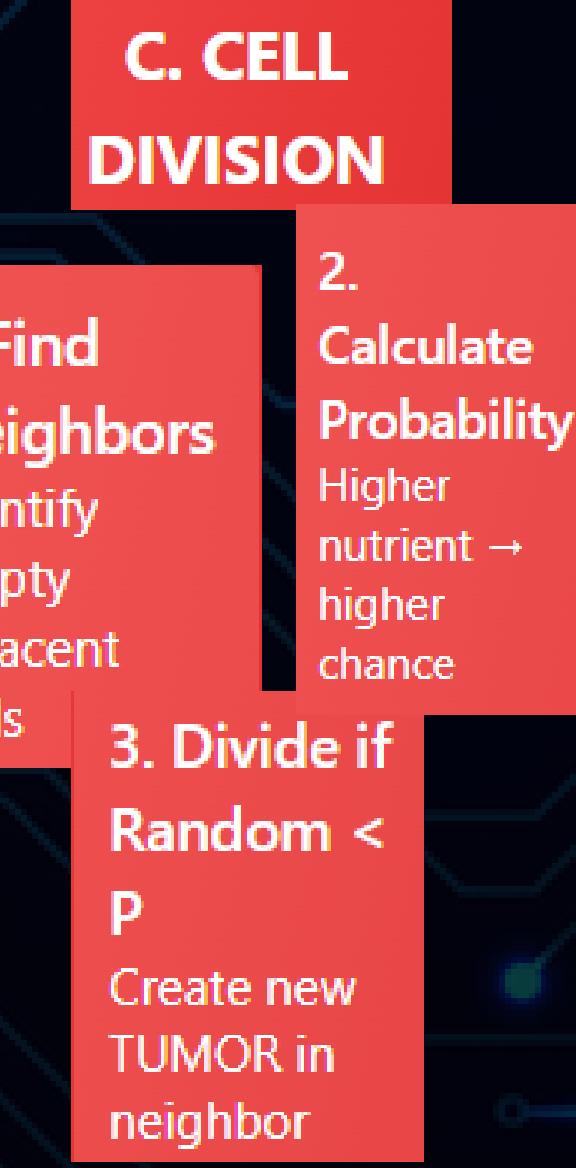
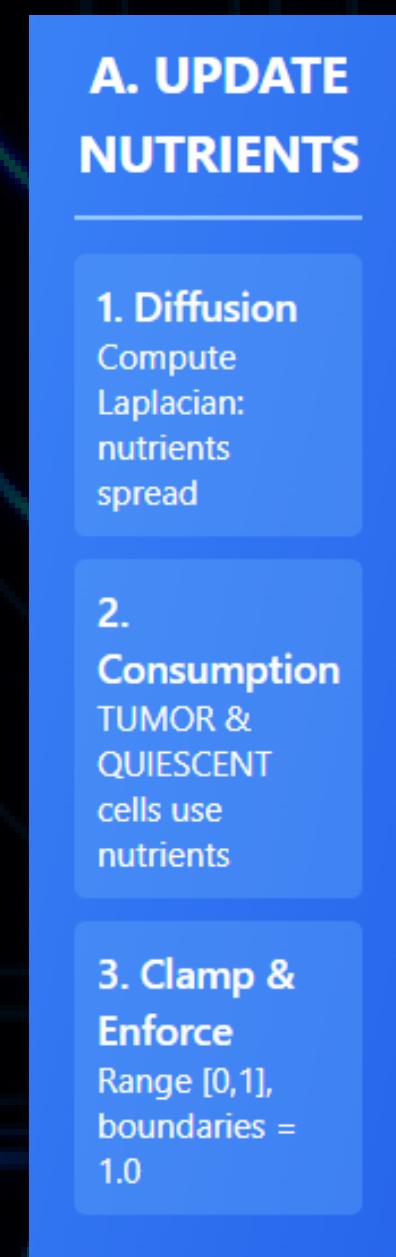
CELL STATES			
Current State	Nutrient Condition	Neighbour Condition (Moore 8-neigh)	Transition / Rule (combined)
Proliferating (P)	High ( $\geq$ division_threshold)	Few tumour neighbours ( $\leq 3$ ) + at least 1 empty neighbour	Remains P / Divides into an empty adjacent cell (space + nutrients allow division)
Proliferating (P)	Moderate $\rightarrow$ Low (< division_threshold but $\geq$ survival_threshold)	Moderate neighbours (4–7)	Becomes Quiescent (Q) due to crowding or reduced nutrients
Proliferating (P)	Very low (< survival_threshold)	Any (usually many neighbours)	Transitions to Q first; then may become N if starvation persists
Quiescent (Q)	High ( $\geq$ division_threshold)	Few tumour neighbours ( $\leq 3$ ) + space available	Reactivates $\rightarrow$ Proliferating (P)
Quiescent (Q)	Moderate (between survival & division thresholds)	Moderate neighbours (4–7)	Stays Quiescent (Q) (not enough nutrients to divide, not low enough to die)
Quiescent (Q)	Very low (< death_threshold)	Fully surrounded (8 tumour neighbours) OR extremely low nutrients	Becomes Necrotic (N) (starvation + crowding)
Necrotic (N)	Any	Any	Remains Necrotic (N) (terminal/absorbing state)

E  
X  
A  
M  
P  
I  
E

LOADING...



# ALGORITHM WORKING



# CODE SNIPPETS

## 4.5 2. Nutrient Diffusion Update

```
1 def update_nutrient_pde(nutrient, grid, D, lam, dt):
2     """Update nutrient field using reaction-diffusion PDE"""
3     # Compute Laplacian for diffusion
4     lap = laplacian(nutrient)
5
6     # Identify consuming cells (tumor + quiescent)
7     consumer = ((grid == TUMOR) | (grid == QUIESCENT)).astype(float)
8     consumption = lam * consumer * nutrient
9
10    # Explicit Euler integration step
11    N_new = nutrient + dt * (D * lap - consumption)
12
13    # Clamp values to valid range
14    N_new = np.clip(N_new, 0.0, 1.0)
15
16    # Enforce boundary supply (blood vessels)
17    N_new[0, :] = 1.0 # Top boundary
18    N_new[-1, :] = 1.0 # Bottom boundary
19    N_new[:, 0] = 1.0 # Left boundary
20    N_new[:, -1] = 1.0 # Right boundary
21
22    return N_new
```

timestep 60

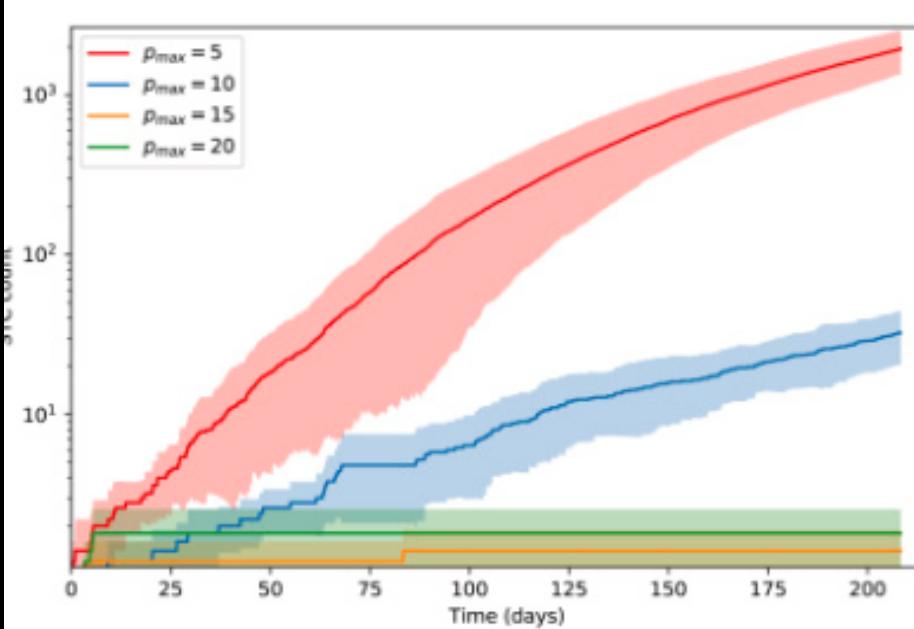


## 4.7 3. Cell Division Logic

```
1 def attempt_divisions(grid, nutrient):
2     """Attempt probabilistic cell division for tumor cells"""
3     births = np.zeros_like(grid, dtype=np.uint8)
4     rng = np.random.default_rng()
5
6     # Get all tumor cell positions
7     tumor_positions = np.argwhere(grid == TUMOR)
8     rng.shuffle(tumor_positions) # Random order
9
10    for (i, j) in tumor_positions:
11        # Find empty neighbor cells (Moore neighborhood)
12        empties = []
13        for di, dj in [(-1,-1), (-1,0), (-1,1), (0,-1), (0,1), (1,-1), (1,0), (1,1)]:
14            x, y = i + di, j + dj
15            if 0 <= x < grid.shape[0] and 0 <= y < grid.shape[1]:
16                if grid[x, y] == EMPTY:
17                    empties.append((x, y))
18
19        if not empties:
20            continue
21
22        # Calculate division probability based on nutrient
23        scaled = (nutrient[i,j] - div_threshold) / (1.0 - div_threshold)
24        prob = p_div * np.clip(base_div_chance + max(0.0, scaled), 0.0, 1.0)
25
26        # Attempt division
27        if rng.random() < prob:
28            x, y = empties[rng.integers(len(empties))]
29            births[x, y] = 1
30
31    return births
```

# Results & Complexity

(a) Evolution of regular tumor cells (RTCs).



## Realistic Patterns

The model replicates the three-zone tumor architecture (proliferating rim, quiescent band, necrotic core). This happens naturally because of nutrient gradients: high nutrients at the edge mean growth, while low nutrients in the center mean dormancy or death.

## Computational Efficiency

- Overall complexity:  $O(T \times N^2)$ .
- Each timestep scans the full  $N \times N$  grid for:
- Nutrient diffusion (Laplacian)
- State transitions
- Cell division attempts
- Each operation is constant-time per cell  $\rightarrow O(N^2)$  per step, repeated  $T$  times.

## Emergent Behavior

Simple local rules (like diffusion, nutrient thresholds, and probabilistic division) can create complex global structures without having to program the shape of the tumor directly.

This causes real-world events like smoothly expanding edges, building up of quiet layers



# LIMITATIONS

## 1. Missing Biological Processes

The model does not include immune cells like T-cells or macrophages.

It uses only one nutrient, while real tumors depend on many things (oxygen, glucose, growth factors). No blood vessel network or angiogenesis (tumor calling blood vessels toward it).

## 2. Grid-Based Simplifications

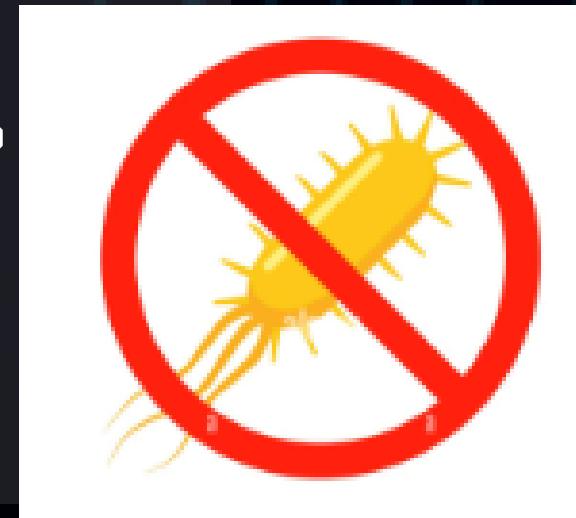
The tumor grows on a square grid, so shapes may look slightly blocky or symmetrical.

Growth may look stronger in straight directions (up/down/left/right) because of grid effects.

## 3. Limited Complexity

No modeling of pH changes, waste buildup, or Warburg effect (real cancer metabolism).

Real tumors have uneven nutrient supply due to complex vessel networks—this model cannot simulate that fully.



# FUTURE SCOPE

## 1. Add More Realistic Biology

Include angiogenesis, so tumors can create new blood vessels.

Add different types of tumor cells using mutations and genetic diversity.

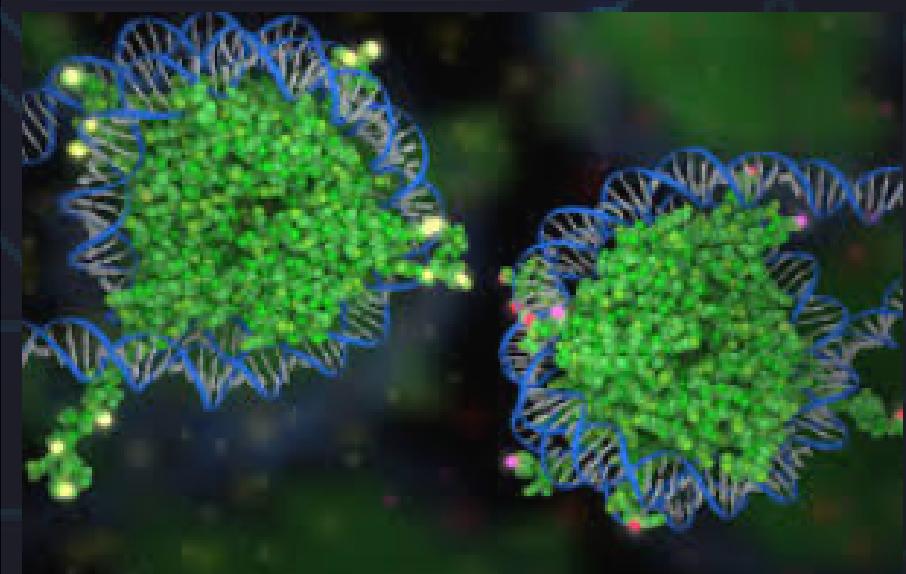
Add immune cells and simulate how they attack or interact with tumors.

## 2. Clinical & Research Applications

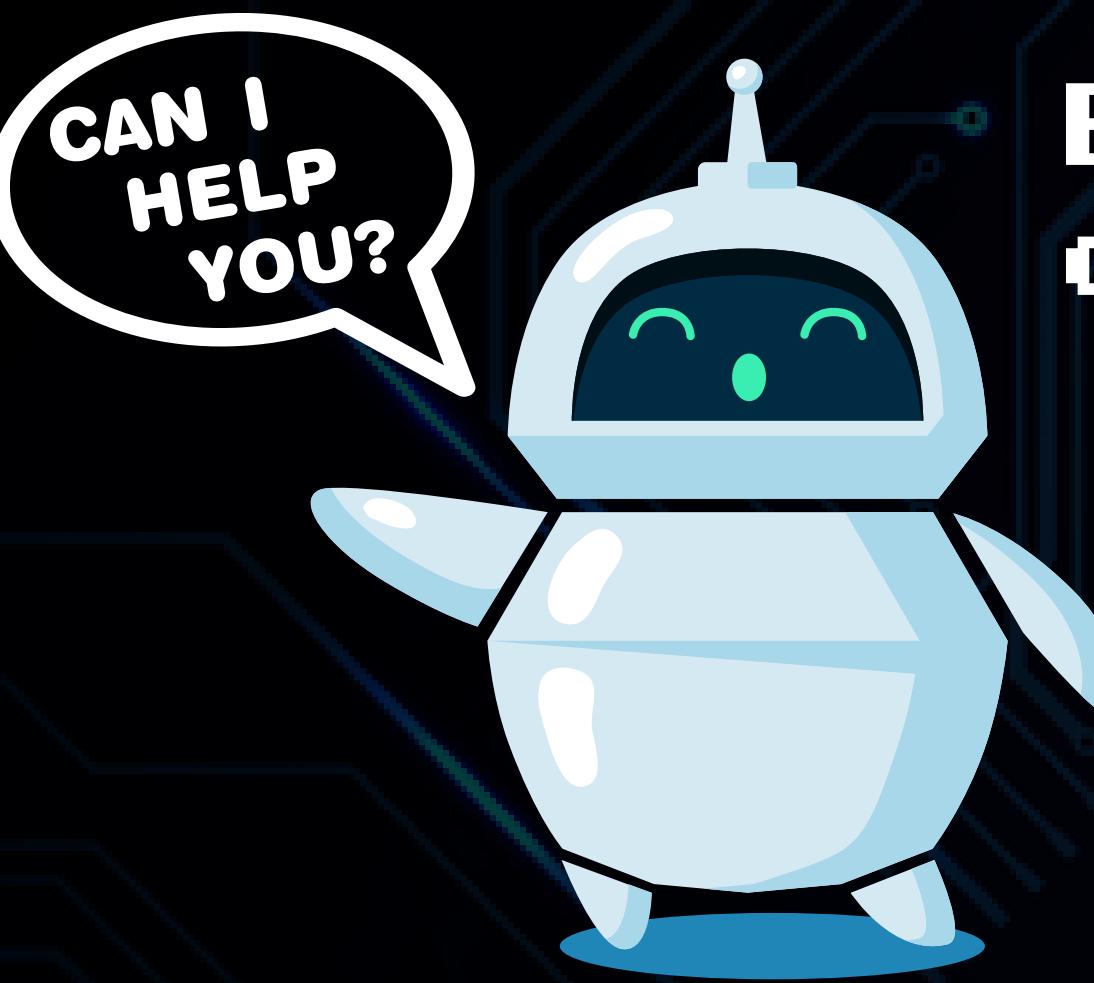
Model how drugs spread inside the tumor and how effective they are.

Use the model to test treatment strategies (e.g., combination therapy, scheduling).

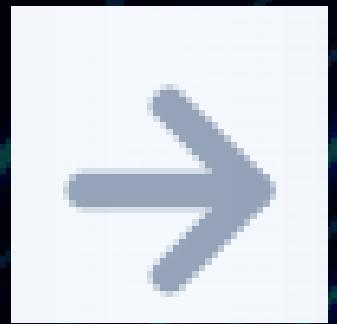
Create patient-specific simulations by tuning parameters to real data, helping with precision medicine.



# THANK YOU !



Enjoy  
our Simulation:



"Any questions? If it's too hard,  
I'll just say 'It's stochastic.'"

timestep 70

