

3D NCA Framework

A Modular and Interactive System for Training, Visualization, and Inference of 3D Neural Cellular Automata

Author: Michal Repčík
Supervisor: Ing. Karel Fritz



Video Demo

Blender UI

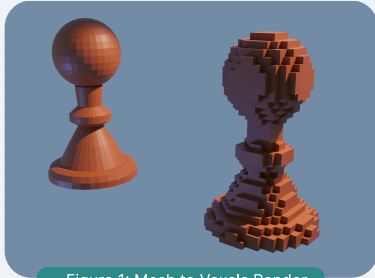


Figure 1: Mesh to Voxels Render

Target Preparation

- Blender mesh → voxelized target
- BSGF or UV texture color
- configurable grid size
- exports .npz for offline use



.npz

3D NCA Framework

- mesh → voxel → training → inference
- Blender ↔ GPU backend over TCP
- local, Colab or Kaggle backend
- no firewall config (ngrok tunnel)

Modular Core

- swap any component independently
 - expand cell vector
 - change perception filters
 - update MLP architecture
 - add new training strategy
- custom runner/trainer = one class
- notebook-first, no Blender needed

Training Settings

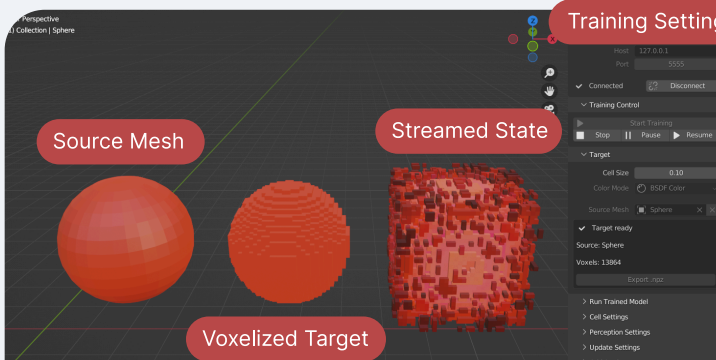


Figure 2: Live Training in Blender

Extensible by Design

- reproduce latest NCA research
 - pool-based morphogenesis
 - multi-target conditional morphogenesis
- add new experiment: subclass NCARunner
- protocol, server, UI unchanged

Blender Viewport

- Configurable training parameters
- Real time training visualization
- Streaming control utilities

TCP Tunnel ↑↓

Colab/Kaggle/Local Backend

Core

Training Pipeline

- Configure NCAModel
 - Cell, Perception, Update
- Configure training parameters
 - LR, optimizer, loss function
- Load training target
- Run, log and export model

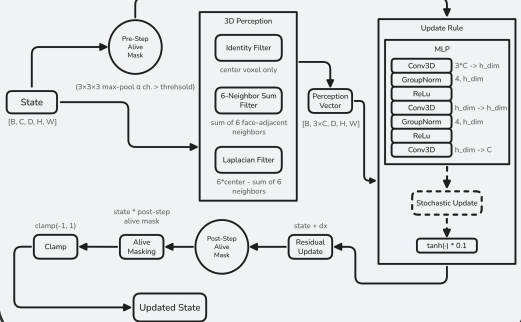


Figure 3: NCA Update Step

Built-in Logging and Visualization

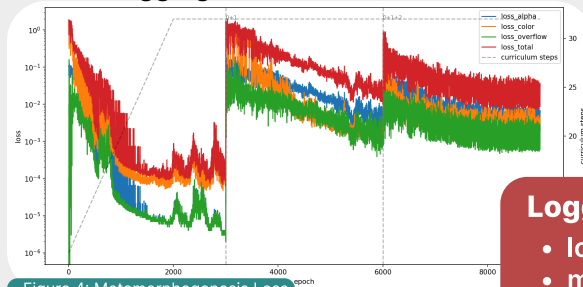
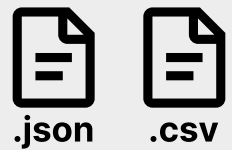


Figure 4: Metamorphogenesis Loss



Logging and Plotting

- loss, metadata, events
- model checkpoints
- plotting utilities

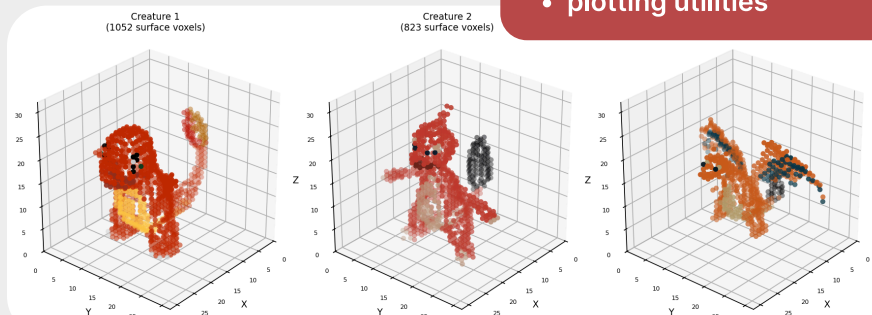


Figure 5: Metamorphogenesis Targets

Visualization utilities

- offline visualization
- matplotlib, pyvista