Neighborhood Prediction in Conway's Game of Life: MLP Baseline Study

Zhang Yiran

November 20, 2025

Abstract

We study how well the current state of a cell in Conway's Game of Life can be predicted from its local spatial neighborhood. Using a simple MLP baseline, we compare 3×3 versus 5×5 neighborhoods (with the center cell masked) across a grid of dynamical regimes, initial densities, and random seeds. The 5×5 model consistently outperforms the 3×3 model, indicating that extended local context carries additional predictive information even in a fully deterministic system.

1 Experimental Setup

1.1 Prediction Task

For a given board state S_t in Conway's Game of Life, we randomly sample valid center positions (x, y) and construct two types of local neighborhood inputs:

- 3×3 patch: we observe the 8 surrounding cells and mask the center,
- 5×5 patch: we observe the 24 surrounding cells and mask the center.

In both cases, the *target* is the true binary state of the center cell at the same time step t. Thus, supervision is identical; the only difference between conditions is the amount of local spatial information provided to the model.

1.2 Data Generation

To explore different dynamical behaviors, we vary:

- **Time regimes** (burn-in steps before sampling):
  - Early: burn-in = 10
  - Mid: burn-in = 60
  - Late: burn-in = 160
- **Initial alive densities**: $p_{\text{alive}} \in \{0.2, 0.4, 0.6\}$
- **Random seeds**: 3 seeds per (regime, density) pair.

For each configuration (regime, density, seed), we generate:

- 16 independent training boards and 4 independent test boards,

- board size: 128×128,
- for each board, 40 time steps are stored after burn-in,
- from each stored step, 200 random valid center positions are sampled.

For every sampled center, we extract both the 3×3 and 5×5 neighborhoods and the corresponding center label. This yields paired datasets under all $3 \times 3 \times 3$ regime–density combinations and 3 seeds, for a total of 27 configuration settings.

1.3 MLP Model

We use a simple MLP architecture to isolate the effect of neighborhood size:

- Input dimension: 8 (for 3×3) or 24 (for 5×5),
- Two hidden layers with 128 ReLU units each,
- Output: single logit for binary classification,
- Optimizer: Adam with learning rate 10^{-3},
- Batch size: 1024,
- Training epochs: 10.

Apart from the input dimensionality, the training protocol and model capacity are kept identical across the two neighborhood conditions.

2 Results

2.1 Overall Performance

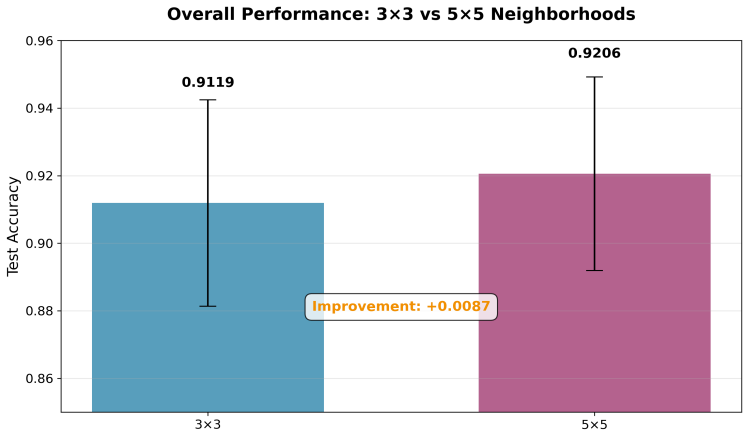

Figure 1: Overall test accuracy of MLP models using 3×3 and 5×5 neighborhoods, averaged over all regime–density–seed combinations. Error bars indicate standard deviations across runs.

Averaged over all 27 configuration settings (time regime, density, seed), the 5×5 neighborhood yields higher test accuracy than 3×3. The mean test accuracies are:

$$\mathrm{ACC}_{3\times3} = 0.9119, \qquad \mathrm{ACC}_{5\times5} = 0.9206,$$

corresponding to an average absolute improvement of

$$\Delta = \mathrm{ACC}_{5\times5} - \mathrm{ACC}_{3\times3} \approx +0.0087.$$

2.2 Effect of Dynamical Regime

Grouping runs by dynamical regime, we obtain mean test accuracies:

$$\begin{aligned} \mathrm{ACC}_{3\times3}^{\mathrm{early}} &= 0.8777, \quad \mathrm{ACC}_{5\times5}^{\mathrm{early}} = 0.8886, \\ \mathrm{ACC}_{3\times3}^{\mathrm{mid}} &= 0.9153, \quad \mathrm{ACC}_{5\times5}^{\mathrm{mid}} = 0.9246, \\ \mathrm{ACC}_{3\times3}^{\mathrm{late}} &= 0.9426, \quad \mathrm{ACC}_{5\times5}^{\mathrm{late}} = 0.9485. \end{aligned}$$

The corresponding improvements are:

$$\Delta_{\mathrm{early}} \approx +0.0109, \quad \Delta_{\mathrm{mid}} \approx +0.0092, \quad \Delta_{\mathrm{late}} \approx +0.0059.$$

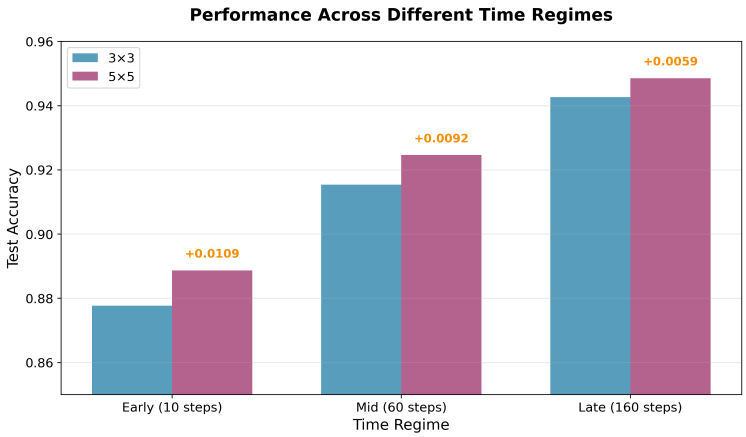

Figure 2: Test accuracy across dynamical regimes (early, mid, late) for 3×3 and 5×5 neighborhoods.

The gain from using 5×5 is present in all three regimes, and is somewhat larger in the early and mid stages where local patterns are more dynamic and heterogeneous.

2.3 Effect of Initial Alive Density

Grouping runs by initial alive density, we obtain:

$$\begin{aligned} \mathrm{ACC}_{3\times3}^{p=0.2} &= 0.9128, \quad \mathrm{ACC}_{5\times5}^{p=0.2} = 0.9218, \\ \mathrm{ACC}_{3\times3}^{p=0.4} &= 0.8991, \quad \mathrm{ACC}_{5\times5}^{p=0.4} = 0.9081, \\ \mathrm{ACC}_{3\times3}^{p=0.6} &= 0.9238, \quad \mathrm{ACC}_{5\times5}^{p=0.6} = 0.9319. \end{aligned}$$

The corresponding improvements are:

$$\Delta_{p=0.2} \approx +0.0089, \quad \Delta_{p=0.4} \approx +0.0090, \quad \Delta_{p=0.6} \approx +0.0081.$$

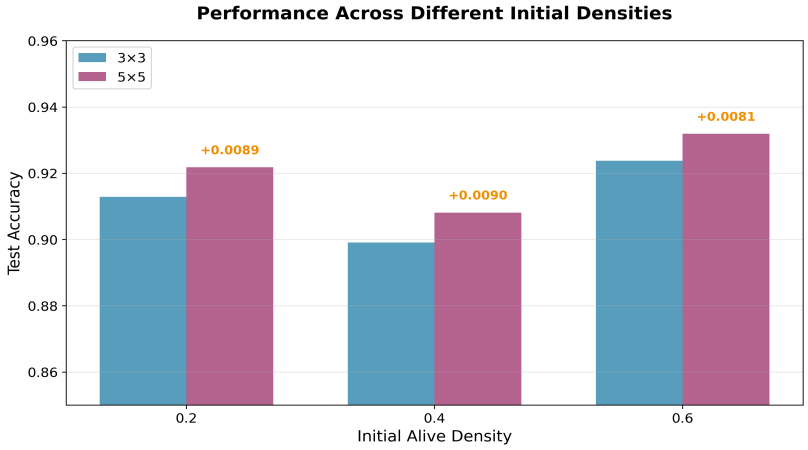

Figure 3: Test accuracy across initial alive densities for 3×3 and 5×5 neighborhoods.

The advantage of 5×5 holds across sparse, medium, and dense initializations, with slightly larger gains at lower densities.

2.4 Joint View: Regime $\times$ Density

To summarize all nine (regime, density) combinations, we compute the accuracy difference $\Delta = \mathrm{ACC}_{5\times5} - \mathrm{ACC}_{3\times3}$ for each pair and visualize it as a heatmap. In all nine cases, Δ is positive and lies between roughly $+0.005$ and $+0.011$.

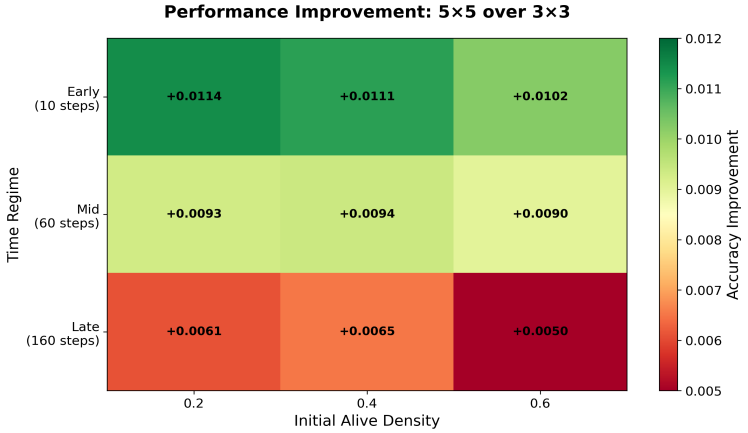

Figure 4: Accuracy improvement of 5×5 over 3×3 neighborhoods across all combinations of dynamical regime and initial density. All entries are positive, indicating consistent benefit from the larger neighborhood.

3 Conclusion

The MLP experiments show that:

- The current state of a cell in Conway's Game of Life can be predicted from local neighborhoods with high accuracy.

- Using a 5×5 neighborhood instead of 3×3 yields a small but consistent improvement (on average +0.0087 absolute accuracy) across all 27 configurations.
- The benefit of larger neighborhoods is present for all time regimes and densities, and tends to be more pronounced when local patterns are more complex (early/mid regimes, sparser boards).

These results indicate that extended local microenvironment contains additional predictive information beyond the immediate neighbors, even in a fully deterministic system.

4 Next Steps: GeoGNN Experiments

In this section we outline how to extend the current MLP-based study to graph neural networks (GeoGNN) on Conway's Game of Life. The goal is to keep the prediction task aligned with the MLP baselines (predicting the current state of a cell from its microenvironment), while exploiting explicit graph structure over local neighborhoods or the full board. Only the experimental design is described here; results will be added after the GeoGNN models are trained.

4.1 Patch-level GeoGNN vs. MLP

The first stage directly parallels the existing MLP experiments and serves as a clean, controlled comparison between *vector-based* and *graph-based* modeling of the same local neighborhood.

Samples. We reuse the existing 5×5 patch dataset: each sample corresponds to a board, a time step t, and a center position (x, y), from which a 5×5 window is extracted. As before, the true center state $S_t(x, y) \in \{0, 1\}$ is used as the binary label.

Graph construction. Instead of flattening the 5×5 window into a 24-dimensional vector (excluding the center), we treat the patch as a small graph:

- Nodes: the 25 cells in the 5×5 window.
- Edges: 8-neighborhood connectivity (up/down/left/right and diagonals) restricted to the patch, i.e. each node is connected to the Game-of-Life neighbors that also lie inside the 5×5 window.
- Node features:
  - For all non-center nodes: the current alive/dead status (0/1).
  - For the center node: a *masked* feature, e.g. a constant value or a learned mask embedding, so that its true state is not directly visible to the model.

Thus, each training sample is a 25-node graph with fixed local connectivity, together with a single binary label for the (masked) center node.

GeoGNN model. We apply a small message-passing GNN on this patch graph, for example:

- 2–3 layers of a standard GNN (e.g. GraphSAGE, GAT, or GIN), with shared parameters across all patches,

- hidden dimension comparable to the MLP baseline (e.g. 64 or 128),
- non-linear activations (ReLU) and optional batch normalization.

After K layers of message passing, we obtain a final embedding $h_{\text{center}}^{(K)}$ for the center node, which is passed through a small readout MLP to produce a logit $\hat{z}$ and probability $\hat{p} = \sigma(\hat{z})$.

Training objective. The loss for a patch sample is the binary cross-entropy between $\hat{p}$ and the true center state $y \in \{0, 1\}$:

$$\mathcal{L}_{\text{patch}} = \text{BCE}(\hat{p}, y).$$

Training is performed over the same train/test splits as the MLP baselines (same boards, time regimes, densities, and seeds). This allows a direct comparison between:

- MLP with 5×5 flattened input (24 features),
- patch-level GeoGNN with the same 5×5 spatial extent, but explicit graph structure.

Planned comparisons. Once trained, we will compare:

- Overall test accuracy of patch-level GeoGNN vs. MLP-5×5,
- Performance across dynamical regimes (early/mid/late),
- Performance across initial alive densities $(0.2, 0.4, 0.6)$.

Because the input region and supervision are identical, any consistent improvement of GeoGNN over MLP can be attributed to the graph-based inductive bias and explicit encoding of neighborhood structure.

4.2 Full-board GeoGNN and Mask–Demask

In a second stage, we move from patch-level graphs to full-board graphs, closer to the mask–demask formulation used in CIFM.

Full-board graph. For a given board state S_t (e.g. 64×64 or 128×128), we construct a grid graph:

- Nodes: one node per cell in the board.
- Edges: 8-neighborhood connectivity on the full grid (each cell connected to its Game-of-Life neighbors).
- Node features: alive/dead status (0/1), optionally augmented with positional encodings.

Masking scheme. Instead of masking only a single center as in the patch-level setting, we adopt a mask–demask strategy:

- At each training iteration, we randomly select a subset of nodes (e.g. 10–30% of all cells on the board),
- For these masked nodes, their input features are replaced by a mask value or embedding; neighbors remain visible.

The GeoGNN then operates on this partially observed graph and attempts to reconstruct the states of the masked nodes.

GeoGNN model and objective. We apply a multi-layer GeoGNN (e.g. 3–5 layers) over the full-board graph. For each masked node i, the model produces a predicted probability $\hat{p}_i$ of being alive. The loss is computed only on the masked nodes:

$$\mathcal{L}_{\text{board}} = \sum_{i \in \mathcal{M}} \text{BCE}\big(\hat{p}_i, y_i\big),$$

where $\mathcal{M}$ is the set of masked nodes and y_i is the true state of cell i at time t.

Planned analyses. After training, we will evaluate:

- Reconstruction accuracy on masked nodes as a function of mask ratio,
- Performance across different dynamical regimes and densities,
- Comparison to local baselines (e.g. MLP-5×5 applied around each node) to quantify the benefit of global message passing.

This full-board mask–demask setup is closer to the original CIFM formulation, where each cell is reconstructed from its microenvironment via a graph-based model.

4.3 Summary of GeoGNN Experimental Goals

In summary, the planned GeoGNN experiments aim to:

- Replace the MLP-5×5 baseline with a patch-level GeoGNN on the same local neighborhood and compare predictive power.
- Extend to full-board graphs with mask–demask training, where multiple cells are masked and reconstructed simultaneously.
- Systematically study how graph structure and neighborhood radius affect the ability to infer current cell states from their microenvironment in a deterministic dynamical system.

The results from these experiments will be added once the GeoGNN models have been implemented and trained.