Component analysis of CNN-based architectures for reverse game of life problem

Ángel F. Caravaca[1]✉[0009−0002−9900−1187], David Guijo-Rubio[1][0000−0002−8035−4057], and Víctor M. Vargas[2][0000−0002−0700−275X]

[1] Departamento de Ciencia de la Computación e Inteligencia Artificial, Universidad de Córdoba, Córdoba, Spain
q02ferca@uco.es, dguijo@uco.es

[2] Departamento de Teoría de la Señal y Comunicaciones, Universidad de Alcalá, Alcalá de Henares, Spain
victor.vargas@uah.es

Abstract. Recent advancements in the reverse Game of Life problem, i.e., finding a predecessor of a given final state, have demonstrated the efficacy of integrating a Differentiable Game of Life layer within convolutional neural networks along with a fuzziness index penalty, i.e., the DiffGoL methodology. However, the exact contribution of each individual component to the overall performance of the model remains unquantified, raising questions about the necessity and impact of each element. In this paper, we present a comprehensive ablation study of the DiffGoL methodology by evaluating the performance of the model under various configurations where specific components are systematically removed. These variants are validated on states of size 15×15, 20×20 and 25×25 after one generation by analysing structure-based metrics along with the Mean Squared Error of the predicted heatmaps of the initial and final states. Validation criteria prioritise evolutionary consistency toward the final states and the structural fidelity of the initial heatmaps directly against the ground-truth, without relying on intermediate binarisation thresholds. Our findings reveal that no single component is solely responsible for the model's performance. Instead, the absence of the DiffGoL layer leads to evolutionary inconsistency, whereas removing the fuzziness index penalty results in severe binarisation failure. This highlights the strict synergy between the architectural components, establishing a robust framework for mapping continuous neural representations to discrete cellular automata spaces.

Keywords: Reverse Game of Life problem · Differentiable Game of Life Layer · Convolutional Neural Networks · Cellular Automata.

1 Introduction

Conway's Game of Life (GoL) [3] is a Cellular Automaton (CA) originally formulated on a two-dimensional grid where cells can be either alive or dead. Specifically, it is a deterministic CA whose evolution depends on the sum of the states in

the Moore neighbourhood [7] of each cell. Let $S_t \in \{0,1\}$ and $N_t \in \{0,1,\ldots,8\}$ be, respectively, the state of a cell at step t and the sum of the states in its Moore neighbourhood. The transition rules are then defined as follows:

$$S_{t+1} = \begin{cases} 1, & \text{if } S_t = 0 \wedge N_t = 3 \\ 1, & \text{if } S_t = 1 \wedge N_t \in \{2,3\} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The Reverse Game of Life (RGoL) problem focuses on finding one of the predecessor states that evolves into a given final state after a certain number of generations. Due to the non-linear and chaotic nature of GoL, the non-injectivity of its global transition function, and the exponentially large search space, this reverse mapping is ambiguous and computationally intractable. To the best of our knowledge, this problem has only been addressed in Kaggle competitions [4]. Building on the solutions of these competitions, [1] proposes a CNN-based approach to address the problem. Specifically, they incorporate a Differentiable Game of Life (DiffGoL) layer into the CNN, together with a fuzziness-index penalty (the DiffGoL methodology). The resulting model predicts initial-state heatmaps that effectively capture the spatial organisation of the ground truth while respecting the underlying dynamics of the automaton, without the need for thresholding.

However, the contribution of each individual component to the overall performance of the DiffGoL methodology remains unquantified. This work therefore presents a comprehensive ablation study, evaluating the model's performance under various configurations in which specific components are systematically removed.

2 Methodology

This section details the methodology proposed to address the Reverse Game of Life (RGoL) problem. We first introduce the Differentiable Game of Life (DiffGoL) transition function, a continuous relaxation of the discrete cellular automata rules. Then, we present a custom loss function designed to enforce spatial fidelity, dynamic coherence, and implicit binarisation of the predictions. Finally, we outline the architectures evaluated in this study.

2.1 DiffGoL transition function

In order to include the GoL dynamics within the model, a differentiable relaxation of the discrete GoL rules that runs over probability maps is proposed in [1]. This is called the Differentiable Game of Life (DiffGoL) and it is built on the DiffGoL rule:

$$g(c) = \frac{1}{1 + \exp\big(\beta(N-c)^2\big)}, \tag{2}$$

which acts as an indicator of the condition $N = c$. Here, N is the sum of the Moore neighbourhood computed using a 3×3 kernel. Moreover, β is a steepness hyperparameter that controls how sharply $g(c)$ peaks at $N = c$. Combining the rule functions for $c = 2$ and $c = 3$, the full transition for a cell with continuous state $x \in [0, 1]$ reads:

$$\Psi(x, N) = 2(1 - x)^n g(3) + 2x^n \big(g(3) + g(2)\big), \tag{3}$$

where $n > 1$ governs the strictness of the relaxation applied to the cell state. The first term represents the birth rule, activating when $x \to 0$ and $N \to 3$, while the second encodes the survival rule, activating when $x \to 1$ and $N \in \{2, 3\}$. For sufficiently large n, intermediate values $x \approx 0.5$ are inherently suppressed, yielding implicit binarisation without explicit thresholding. Moreover, in the binary limit, Ψ exactly reproduces the discrete GoL transition. Figure 1 illustrates the landscape of Ψ for $\beta = 50$ and $n = 4$.

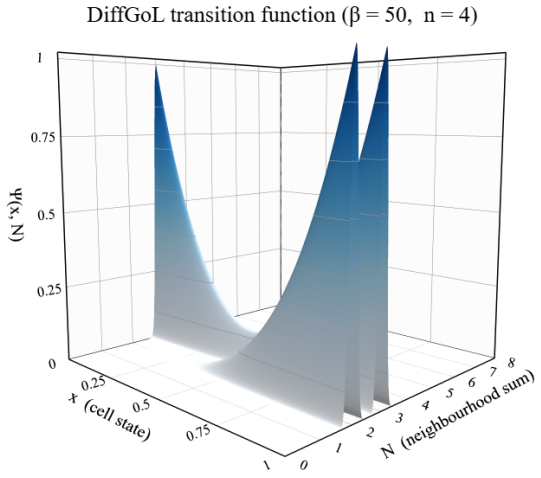


Fig. 1. Differentiable Game of Life transition function Ψ for $\beta = 50$ and $n = 4$. The function is plotted as a function of the cell state x and the neighbourhood sum N. The transition function exhibits sharp peaks at $N \in \{2, 3\}$, with decay away from these values, effectively capturing the dynamics of GoL while maintaining differentiability.

2.2 Fuzziness index and loss function

The validation of the RGoL problem methodologies should reflect the intrinsic properties of GoL. First, since GoL dynamics are highly sensitive to the spatial arrangement of cells, a meaningful evaluation must assess the structural fidelity of the predicted initial state with respect to the ground truth. Second, the predicted state must be coherent with the GoL transition function, i.e., it

must evolve into the given final state under the GoL dynamics. Third, given the binary nature of GoL, predicted boards should be as binary as possible prior to any post-processing step, so that binarisation introduces minimal error and the resulting board is a valid GoL state with the highest possible confidence. To achieve this, a fuzziness index penalty is introduced to force the model to produce more binarised predictions. For $H \times W$ states, this index is defined as:

$$\Phi(\hat{y}) \;=\; \frac{4}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \hat{y}_{ij}\,(1 - \hat{y}_{ij}), \tag{4}$$

where $\hat{y} \in [0, 1]$ is the predicted initial state heatmap. Notice that the term $\hat{y}_{ij}\,(1 - \hat{y}_{ij})$ is maximised at $\hat{y}_{ij} = 0.5$ and minimised at $\hat{y}_{ij} \in \{0, 1\}$, thus penalising intermediate values. The factor of 4 normalises the index to the range $[0, 1]$.

Furthermore, the DiffGoL layer enables a loss function that combines a spatial fidelity term with a GoL dynamics consistency term. The loss is built as a weighted sum of three components: the Focal Binary Cross-Entropy (FBCE) [5] between the predicted and ground-truth initial states $(\hat{y}_0, y_0)$, the Binary Cross-Entropy (BCE) [2] between the DiffGoL-evolved predicted initial state and the final state $(\hat{y}_f, y_f)$, and the fuzziness index penalty applied to the predicted initial states. This is expressed as:

$$\mathcal{L}(\hat{y}_0, y_0, \hat{y}_f, y_f) \;=\; \lambda_1\, \mathcal{L}_{\text{FBCE}}(\hat{y}_0, y_0) \;+\; \lambda_2\, \mathcal{L}_{\text{BCE}}(\hat{y}_f, y_f) \;+\; \lambda_3\, \Phi(\hat{y}_0). \tag{5}$$

Here, λ_1, λ_2 and λ_3 are hyperparameters that balance the contribution of each term, and are systematically varied throughout this ablation to assess the impact of each component of the methodology on the overall performance.

2.3 Model's architecture

In [1], three CNN-based architectures are proposed to address the RGoL problem. The base architecture, denoted as the *Classic Model*, is a standard CNN formed by stacking blocks of convolutional layers and batch normalisation with ReLU activations, and ends with a final convolutional layer with sigmoid activation that produces the predicted initial state heatmap. The second and third architectures incorporate the DiffGoL layer after this final convolutional layer, and are denoted as the *DiffGoL Model* and the *Annealed-DiffGoL Model*, respectively. The latter differs from the former in that it applies an annealing schedule to the hyperparameters λ_2 and λ_3 of the loss function. Since the Annealed-DiffGoL Model has been shown to yield no significant performance gain over the DiffGoL Model [1], we restrict this ablation to the Classic and DiffGoL models, focusing on the contribution of the DiffGoL layer and the fuzziness index penalty to the overall performance. The architectures of these models are illustrated in Figure 2. Lastly, batch normalisation layers are critical, as they stabilise and smooth the training process, enabling to use higher β and n values in the DiffGoL layer, which yields sharper transitions and more binarised predictions.

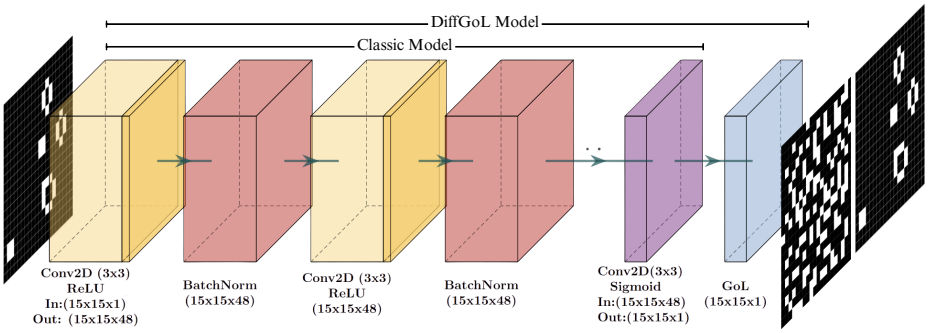


Fig. 2. Architectures of the Classic and DiffGoL models. The Classic Model consists of a series of convolutional blocks followed by a final convolutional layer with sigmoid activation. The DiffGoL Model incorporates the DiffGoL layer (blue block) after the final convolutional layer.

2.4 Evaluation metrics

As discussed in [1], limiting the evaluation of RGoL methodologies to pixel-wise metrics such as those related to the confusion matrix yields misleading results, since GoL is a CA whose evolution is determined by local interactions, and thus the spatial organisation of the predicted initial state is critical. Therefore, we evaluate the performance of the models using a combination of structure-based metrics and the Mean Squared Error (MSE) on the predicted heatmaps of the initial and final states.

The structure-based metrics include the Earth Mover's Distance (EMD)[6], which quantifies the minimum cost of transforming the predicted initial state into the ground truth. Furthermore, the degree of binarisation of the predicted initial states is measured by comparing their living cell density to that of the ground truth, and by computing their fuzziness index and the proportion of cells whose predictions fall within a High-Uncertainty Range (%HUC). For instance, a $\%\text{HUC}_{[0.45,\,0.55]}$ of 75% for an $H \times W$ state indicates that 75% of its cells are predicted with a probability in the range $[0.45, 0.55]$, reflecting a highly uncertain prediction.

3 Experimental settings

This section describes the experimental settings used in our ablation study of the DiffGoL methodology. In contrast to [1], where the Classic, DiffGoL and Annealed-DiffGoL models are compared using the TensorFlow framework, in this work we focus on a PyTorch implementation of the Classic and DiffGoL models. The Annealed-DiffGoL model is omitted because it showed similar performance to the DiffGoL model in the original paper. Moreover, the switch to PyTorch is motivated by the fact that, compared to TensorFlow, PyTorch's `conv2d` natively

supports periodic padding, which allows us to implement the GoL boundary conditions directly during the convolution operations. This is particularly relevant for achieving predictions that are more structurally consistent with the GoL dynamics, as we will see in the next section. For reproducibility, both versions of the code are publicly available at the following GitHub repository[3].

3.1 Training and test pipelines

The training and test pipelines are implemented in PyTorch, following the same data generation process as in [1]. The main steps can be summarised as follows.

Data generation: first, we generate a random set of GoL configurations with densities distributed in the range $[0.05, 0.95]$. To achieve stability in their evolution, we let them evolve for 3 warm-up steps, yielding the candidate initial states. Then, we evolve these states for one additional step to obtain the corresponding final states. Finally, we retain only those pairs whose density falls within the previously mentioned range.

Training: the training process consists of 1000 epochs, in which 60,000 training samples are generated on the fly according to the validation loss. Specifically, if the validation loss does not improve for 25 consecutive epochs, a new set of training samples is generated. If, before this threshold is reached, the validation loss does not improve for 20 consecutive epochs, a reduce-on-plateau scheduler is applied, halving the learning rate. This strategy is designed to prevent overfitting while avoiding the cost of continually regenerating the data. The model is saved at every epoch in which the validation loss improves with respect the historical best.

Testing: to validate the methodology, the mean and standard deviation of the metrics described in the previous sections are computed over a single fixed test dataset across 15 models trained with different random seeds. This allows us to assess the consistency of the results across different weights initialisations.

3.2 Ablation study

In order to assess the impact of each component of the DiffGoL methodology, we perform an ablation study by comparing the performance of the models under various configurations. These are defined by the presence or absence of the FBCE loss on the initial states, the DiffGoL layer, the Fuzziness Index penalty, or any combination of them. In particular, the absence of the DiffGoL layer allows us to evaluate the importance of including the GoL dynamics in the architecture. The absence of the Fuzziness Index penalty allows us to evaluate the capability of the model to inherently predict more binarised initial states. In addition, since the inverse GoL problem is non-injective, to analyse the effects of supervising

[3] `https://github.com/ayrna/reverseGoLwithCNN`

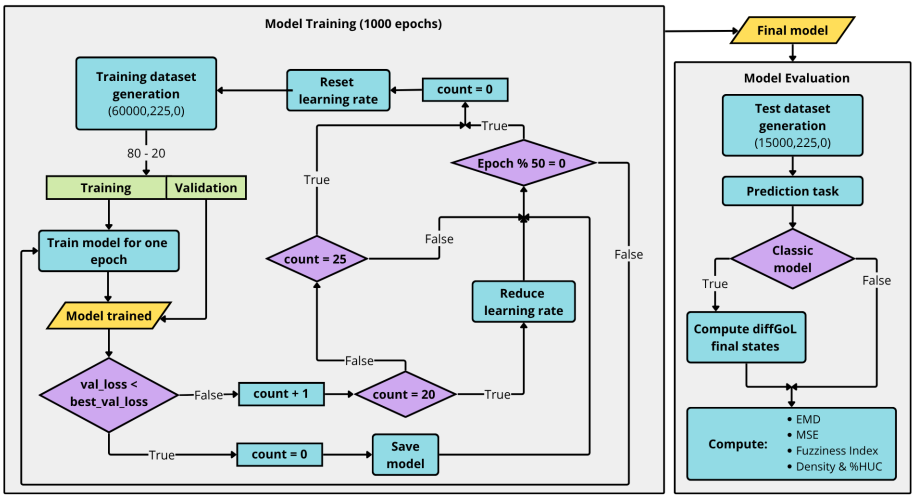


Fig. 3. Training and testing pipeline of the Classic and DiffGoL models.

directly on the initial states, we also ablate the FBCE loss between predicted and target initial states. This allows us to evaluate whether training the model with specific initial states is necessary or whether the model can recover valid states guided only by the forward dynamics.

To control the presence or absence of these components we use the λ_i parameters defined in the loss function (Equation (5)). In particular, λ_1 controls the FBCE loss between the predicted and target initial states, while λ_2 controls the presence of the DiffGoL layer, as it computes the BCE loss between the DiffGoL-evolved predicted initial states and the final states. On the other hand, λ_3 is the weight of the Fuzziness Index penalty, thus controlling its presence in the architecture. We will refer to these configurations as *Model* $\lambda_1\lambda_2\lambda_3$, yielding 6 meaningful configurations out of the 8 possible combinations. *Model 000* is excluded as it provides no learning signal, while *Model 001* is omitted because the Fuzziness Index penalty alone does not relate the initial state prediction to the target. Moreover, *Model 100* corresponds to the Classic model, while *Model 111* corresponds to the DiffGoL model. A summary of the configurations is provided in Table 1.

4 Results and discussion

In this section, we present the experimental evaluation of the DiffGoL methodology. First, we compare the TensorFlow and PyTorch implementations of both the Classic and DiffGoL models, highlighting the impact of incorporating the GoL boundary conditions into the convolution operations. Then, the results of the ablation study that assesses the contribution of each component of the DiffGoL

Table 1. Summary of the configurations considered in the ablation study. Each row indicates which loss components are active ($\lambda_i = 1$) or removed ($\lambda_i = 0$): λ_1 weights the FBCE on the predicted initial states, λ_2 the BCE on the DiffGoL-evolved final states, and λ_3 the Fuzziness Index penalty.

Configuration	$\boldsymbol{\lambda_1}$ **(Initial)**	$\boldsymbol{\lambda_2}$ **(DiffGoL)**	$\boldsymbol{\lambda_3}$ **(Fuzziness)**
Classic Model (100)	1	0	0
DiffGoL Model (111)	1	1	1
Model 010	0	1	0
Model 011	0	1	1
Model 101	1	0	1
Model 110	1	1	0

methodology is included. To provide a comprehensive evaluation, all performance metrics are reported as mean ± standard deviation over 15 independent seeds, computed on both the initial and the final states.

4.1 Comparison of TensorFlow and PyTorch implementations

The key difference between these implementations is the use of periodic padding in the convolution operations of the PyTorch versions. This allows us to directly incorporate the GoL boundary conditions into the architecture, which is not possible in the TensorFlow implementation without additional modifications. Tables 2 and 3 show the performance metrics for both implementations of the classic and DiffGoL models.

Table 2. Performance comparison of the Classic and DiffGoL models on the initial states across the TensorFlow and PyTorch implementations. Values are reported as mean with standard deviation as subscript across 15 independent seeds. Best results per row are highlighted in **bold** and second best in *italics*. The ground truth density is $22.39_{\pm 9.39}\%$.

		TensorFlow		**PyTorch**	
Metric		**Classic**	**DiffGoL**	**Classic**	**DiffGoL**
Fuzziness Index ↓		$0.7495_{\pm 0.0062}$	*0.3225*$_{\pm 0.0014}$	$0.7471_{\pm 0.1483}$	**0.2809**$_{\pm 0.1455}$
Density (%)		$32.25_{\pm 8.36}$	**23.53**$_{\pm 9.32}$	$32.22_{\pm 8.38}$	*23.84*$_{\pm 9.65}$
EMD ↓		$2.0257_{\pm 0.0227}$	*1.5179*$_{\pm 0.0048}$	$2.0215_{\pm 0.0071}$	**1.3402**$_{\pm 0.0026}$
MSE ↓		$0.1220_{\pm 0.0007}$	$0.1248_{\pm 0.0003}$	*0.1214*$_{\pm 0.0542}$	**0.1107**$_{\pm 0.0650}$
%HUC ↓	[0.45, 0.55]	$12.44_{\pm 0.15}$	*2.76*$_{\pm 0.06}$	$12.31_{\pm 0.10}$	**2.37**$_{\pm 0.05}$
	[0.40, 0.60]	$24.21_{\pm 0.24}$	*5.67*$_{\pm 0.12}$	$23.98_{\pm 0.20}$	**4.86**$_{\pm 0.09}$
	[0.35, 0.65]	$35.46_{\pm 0.31}$	*8.90*$_{\pm 0.17}$	$35.13_{\pm 0.27}$	**7.60**$_{\pm 0.13}$

Analysing the results, we observe that the Classic models perform comparably across both frameworks, as expected given that they share the same ar-

chitecture and differ only in the underlying library. The DiffGoL models, in contrast, show a clear advantage in the PyTorch implementation. This is particularly evident in the EMD, where the PyTorch version substantially outperforms the TensorFlow one, with a relative improvement of approximately 12% on the initial states and, even more notably, of approximately 34% on the final states.

This behaviour indicates that the periodic padding incorporated into the PyTorch convolutions allows the model to better capture not only the structure of the initial states but also the underlying GoL dynamics, which is essential for predicting physically consistent initial states that evolve into the final states. Therefore, we conclude that explicitly incorporating the CA boundary conditions into the architecture is a key factor in the performance of the DiffGoL model.

Table 3. Performance comparison of the Classic and DiffGoL models on the final states across the TensorFlow and PyTorch implementations. Values are reported as mean with standard deviation as subscript across 15 independent seeds. Best results per row are highlighted in **bold** and second best in *italics*.

	TensorFlow		**PyTorch**	
Metric	**Classic**	**DiffGoL**	**Classic**	**DiffGoL**
Fuzziness Index ↓	$\mathbf{0.0696}_{\pm 0.0010}$	$0.1771_{\pm 0.0004}$	$\mathit{0.0696}_{\pm 0.0285}$	$0.1778_{\pm 0.0916}$
EMD ↓	$2.9835_{\pm 0.0328}$	$\mathit{2.0843}_{\pm 0.0082}$	$2.8374_{\pm 0.0346}$	$\mathbf{1.3814}_{\pm 0.0059}$
MSE ↓	$0.2079_{\pm 0.0012}$	$\mathit{0.1013}_{\pm 0.0008}$	$0.2078_{\pm 0.1007}$	$\mathbf{0.0617}_{\pm 0.0401}$

4.2 Ablation study results

To analyse the results obtained in this study, we must discuss the results for the initial and final states jointly, since the ablation is informative only when initial state fidelity and forward consistency are considered together. While our discussion centers on the 15×15 boards (Tables 4 and 5), the observed trends generalize consistently to the 20×20 and 25×25 configurations (Tables 6 and 7). We recall that the evaluation prioritises both the evolutionary consistency of the predictions with the final states and the structural fidelity of the raw predicted heatmaps to the ground-truth initial states, without relying on any intermediate binarisation step. We discuss each configuration in turn.

Classic Model: Trained with the FBCE loss on the initial states alone (λ_1 only), the Classic model behaves as expected for a configuration without binarisation penalty or forward-dynamics supervision. On the initial states (Table 4), it produces the highest Fuzziness Index (≈ 0.75) and the highest %HUC values, with 35% of the predicted pixels within the $[0.35, 0.65]$ uncertainty band, confirming grey rather than binary predictions. The predictions are also structurally inaccurate, with a density of 32.22% far from the ground-truth value of 22.39% and an EMD of ≈ 2.0 indicating spatial displacement. These deficiencies propagate forward, since the fuzzy initial states collapse to empty boards, evidenced

by the final-state MSE (0.208) being close to the empty-board baseline (0.220). The Classic Model therefore fails on both evaluation criteria.

Model 010: Trained solely with forward-dynamics supervision on the final states (λ_2 only), Model 010 exhibits the behaviour expected of a model lacking guidance on the initial states. On the initial states (Table 4), it fails to capture their structure, producing the highest EMD (≈ 3.08) and density (56.50%). On the final states (Table 5), however, the DiffGoL-evolved predictions are consistent with the GoL dynamics, yielding the best MSE (0.028) and the second-best Fuzziness Index (≈ 0.08), with a relative improvement of $\approx 46\%$ in EMD over the Classic model baseline. This is explained by the fact that, since this model lacks initial state supervision, it focuses on finding a predecessor with a consistent evolution rather than a structurally faithful one. Owing to the continuous nature of the DiffGoL layer, the model predicts fuzzy initial states that, despite their high Fuzziness Index, nonetheless evolve to the correct final state. Therefore, although the DiffGoL layer enables the model to predict initial states whose evolution is consistent with the GoL dynamics, initial-state supervision remains necessary to predict structurally faithful initial states that also evolve consistently.

Model 011: This model is trained with the forward-dynamics supervision on the final states and the binarisation penalty on the predicted initial states (λ_2 and λ_3 only). On the initial states (Table 4), it produces the best Fuzziness Index (≈ 0.14) and the second-best %HUC results, although it fails to capture the density of the states (47.12%, more than double the ground truth). Furthermore, since it lacks supervision on the initial states, its EMD (≈ 2.76) and MSE (≈ 0.54) are the second-worst. On the final states (Table 5), it produces the best EMD (≈ 1.19) and the second-best MSE (≈ 0.04), with a low Fuzziness Index (≈ 0.16) indicating binarised DiffGoL-evolved states. Therefore, the binarisation penalty along with the DiffGoL layer allows the model to predict binarised initial states whose evolution is consistent with the GoL dynamics, although they are not structurally consistent.

Model 101: This model is trained with supervision only on the initial states and guidance toward more binarised predictions (λ_1 and λ_3 only). On the initial states (Table 4), it yields good performance across all metrics, achieving the best MSE (≈ 0.11) and the best %HUC results, with the second-best values in Fuzziness Index, Density and EMD. However, on the final states (Table 5), the model fails despite this initial performance. This is evidenced by the second-worst EMD (≈ 2.75), together with a low Fuzziness Index (≈ 0.09) and an MSE (≈ 0.17) close to the empty-board baseline (0.220), indicating that the DiffGoL-evolved states are mostly dead. Therefore, the binarisation penalty and the supervision on the initial states are crucial for structurally accurate initial states, although the lack of the DiffGoL layer leads to initial states whose evolution is not consistent with the GoL dynamics.

Table 4. Ablation study on 15×15 initial states. Values are reported as mean with standard deviation as subscript over 15 seeds. Best and second-best results are shown in **bold** and *italics*, respectively. Ground-truth density: $22.39_{\pm 9.39}\%$.

					%HUC↓		
	Fuzziness Index ↓	**Density (%)**	**EMD ↓**	**MSE ↓**	**[0.45, 0.55]**	**[0.40, 0.60]**	**[0.35, 0.65]**
Classic	$0.7471_{\pm 0.1483}$	$32.22_{\pm 8.38}$	$2.0215_{\pm 0.0071}$	$0.1214_{\pm 0.0542}$	$12.31_{\pm 0.10}$	$23.98_{\pm 0.20}$	$35.13_{\pm 0.27}$
DiffGoL	$0.2809_{\pm 0.1455}$	$\mathbf{23.84}_{\pm 9.66}$	$\mathbf{1.3402}_{\pm 0.0026}$	$\mathit{0.1107}_{\pm 0.0650}$	$2.37_{\pm 0.05}$	$4.86_{\pm 0.09}$	$7.60_{\pm 0.13}$
Model 010	$0.4508_{\pm 0.1686}$	$56.50_{\pm 13.93}$	$3.0790_{\pm 0.1578}$	$0.5454_{\pm 0.1478}$	$9.79_{\pm 3.44}$	$19.25_{\pm 6.81}$	$27.93_{\pm 10.13}$
Model 011	$\mathbf{0.1425}_{\pm 0.0772}$	$47.12_{\pm 12.80}$	$2.7643_{\pm 0.5749}$	$0.5367_{\pm 0.1898}$	$\mathit{0.95}_{\pm 0.04}$	$\mathit{1.94}_{\pm 0.08}$	$\mathit{3.05}_{\pm 0.12}$
Model 101	$\mathit{0.2077}_{\pm 0.1160}$	$\mathit{19.64}_{\pm 7.77}$	$\mathit{1.5123}_{\pm 0.0042}$	$\mathbf{0.1097}_{\pm 0.0666}$	$\mathbf{0.90}_{\pm 0.05}$	$\mathbf{1.87}_{\pm 0.11}$	$\mathbf{3.04}_{\pm 0.16}$
Model 110	$0.5572_{\pm 0.1108}$	$28.46_{\pm 7.34}$	$1.9121_{\pm 0.0326}$	$0.1187_{\pm 0.0592}$	$4.56_{\pm 0.07}$	$9.52_{\pm 0.11}$	$15.05_{\pm 0.15}$

Table 5. Ablation study on 15×15 final states. Values are reported as mean with standard deviation as subscript over 15 seeds. Best and second-best results are shown in **bold** and *italics*, respectively. Reference MSE corresponds to predicting empty states: $0.2200_{\pm 0.0981}$.

Configuration	**Fuzziness Index ↓**	**EMD ↓**	**MSE ↓**
Classic	$\mathbf{0.0696}_{\pm 0.0285}$	$2.8374_{\pm 0.0346}$	$0.2078_{\pm 0.1007}$
DiffGoL	$0.1778_{\pm 0.0916}$	$1.3814_{\pm 0.0059}$	$0.0617_{\pm 0.0401}$
Model 010	$\mathit{0.0845}_{\pm 0.0783}$	$1.5333_{\pm 2.9252}$	$\mathbf{0.0281}_{\pm 0.0595}$
Model 011	$0.1589_{\pm 0.0873}$	$\mathbf{1.1894}_{\pm 0.0170}$	$\mathit{0.0397}_{\pm 0.0287}$
Model 101	$0.0876_{\pm 0.0402}$	$2.7478_{\pm 0.0125}$	$0.1659_{\pm 0.0886}$
Model 110	$0.1958_{\pm 0.0993}$	$\mathit{1.2771}_{\pm 0.0048}$	$0.0620_{\pm 0.0419}$

Model 110: Trained with supervision on both the initial and final states without guidance towards more binarised predictions (λ_1 and λ_2 only), Model 110 behaves similarly to the Classic Model on the initial states, with comparable results across all metrics. However, there is an improvement in the Fuzziness Index (from ≈ 0.75 to ≈ 0.56) and in the %HUC results, due to the smooth binarisation introduced by the DiffGoL layer, although these two metrics still indicate fuzzy initial states. On the final states, the performance is similar to Model 011, yielding consistent DiffGoL-evolved final states with the second-best EMD (≈ 1.28) and an MSE (≈ 0.06) close to that of the full DiffGoL Model. Therefore, the DiffGoL layer is crucial for good performance on the final states, while the lack of the Fuzziness Index penalty leads to structurally inconsistent initial states.

DiffGoL Model: Trained with all three loss components (λ_1, λ_2 and λ_3), it represents the full methodology proposed in [1]. On the initial states (Table 4), it achieves the best density of 23.84%, closest to the ground truth of 22.39%, and the best EMD (≈ 1.34), with the second-best MSE (≈ 0.11) and competitive Fuzziness Index (≈ 0.28) and %HUC values. On the final states (Table 5), it performs comparably to Model 011 and Model 110, with an EMD of ≈ 1.38

and an MSE of ≈ 0.06. Therefore, the DiffGoL Model is the only configuration that achieves balanced performance across both evaluation criteria, producing structurally consistent initial states whose evolution is physically consistent with the GoL dynamics, and thus the best-performing configuration of this ablation study.

Finally, the DiffGoL methodology fails to converge on tasks where the number of generations between the initial and final states is greater. Consequently, the only viable approach to reproduce RGoL dynamics would be the recursive application of a DiffGoL model trained for a single step. This limitation is theoretically expected as the chaotic nature of GoL dynamics causes inherent information loss regarding the initial states over time. The DiffGoL layer directly transfers this behaviour into the learning process, preventing model convergence.

Table 6. Ablation study on initial states across different board sizes. Values are reported as mean with standard deviation as subscript over 15 seeds. Best and second-best results per resolution are shown in **bold** and *italics*, respectively. Ground-truth densities: $21.40_{\pm 8.99}\%$ (20×20) and $20.71_{\pm 8.76}\%$ (25×25).

					%HUC↓		
Configuration	**Fuzziness Index ↓**	**Density (%)**	**EMD ↓**	**MSE ↓**	**[0.45, 0.55]**	**[0.40, 0.60]**	**[0.35, 0.65]**
20×20 Boards							
Classic	$0.7217_{\pm 0.1462}$	$31.11_{\pm 8.24}$	$3.1435_{\pm 0.0072}$	$0.2115_{\pm 0.0397}$	$10.90_{\pm 0.10}$	$21.35_{\pm 0.19}$	$31.53_{\pm 0.24}$
DiffGoL	$0.2653_{\pm 0.1373}$	$\mathbf{22.78}_{\pm 9.28}$	$\mathbf{1.4229}_{\pm 0.0025}$	$\mathit{0.1045}_{\pm 0.0594}$	$2.21_{\pm 0.04}$	$4.53_{\pm 0.08}$	$7.08_{\pm 0.10}$
Model 010	$0.4430_{\pm 0.1672}$	$57.45_{\pm 13.09}$	$3.4836_{\pm 0.2073}$	$0.5522_{\pm 0.1446}$	$9.81_{\pm 3.90}$	$19.19_{\pm 7.31}$	$27.70_{\pm 10.59}$
Model 011	$\mathbf{0.1324}_{\pm 0.0703}$	$45.68_{\pm 12.03}$	$3.0122_{\pm 0.7689}$	$0.5218_{\pm 0.2036}$	$\mathbf{0.87}_{\pm 0.05}$	$\mathbf{1.79}_{\pm 0.09}$	$\mathbf{2.81}_{\pm 0.15}$
Model 101	$\mathit{0.1906}_{\pm 0.1080}$	$\mathit{19.04}_{\pm 7.57}$	$\mathit{1.5861}_{\pm 0.0052}$	$\mathbf{0.0989}_{\pm 0.0588}$	$\mathit{1.00}_{\pm 0.02}$	$\mathit{2.06}_{\pm 0.04}$	$\mathit{3.28}_{\pm 0.06}$
Model 110	$0.5495_{\pm 0.1040}$	$27.72_{\pm 7.08}$	$2.0909_{\pm 0.0385}$	$0.1132_{\pm 0.0552}$	$4.31_{\pm 0.06}$	$9.02_{\pm 0.11}$	$14.29_{\pm 0.17}$
25×25 Boards							
Classic	$0.7041_{\pm 0.1435}$	$30.29_{\pm 8.08}$	$3.4431_{\pm 0.0107}$	$0.2085_{\pm 0.0399}$	$9.97_{\pm 0.09}$	$19.60_{\pm 0.16}$	$29.09_{\pm 0.21}$
DiffGoL	$0.2547_{\pm 0.1321}$	$\mathbf{22.05}_{\pm 9.05}$	$\mathbf{1.4887}_{\pm 0.0032}$	$\mathit{0.1003}_{\pm 0.0563}$	$2.12_{\pm 0.03}$	$4.34_{\pm 0.06}$	$6.79_{\pm 0.07}$
Model 010	$0.4341_{\pm 0.1700}$	$59.02_{\pm 13.33}$	$3.8374_{\pm 0.2227}$	$0.5673_{\pm 0.1461}$	$9.63_{\pm 4.03}$	$18.82_{\pm 7.43}$	$27.19_{\pm 10.70}$
Model 011	$\mathbf{0.1260}_{\pm 0.0663}$	$48.50_{\pm 12.46}$	$3.4028_{\pm 0.8362}$	$0.5473_{\pm 0.2011}$	$\mathbf{0.79}_{\pm 0.05}$	$\mathbf{1.63}_{\pm 0.11}$	$\mathbf{2.57}_{\pm 0.17}$
Model 101	$\mathit{0.1785}_{\pm 0.1024}$	$\mathit{18.56}_{\pm 7.44}$	$\mathit{1.6444}_{\pm 0.0056}$	$\mathbf{0.0921}_{\pm 0.0544}$	$\mathit{1.01}_{\pm 0.03}$	$\mathit{2.07}_{\pm 0.06}$	$\mathit{3.27}_{\pm 0.09}$
Model 110	$0.5445_{\pm 0.0978}$	$27.25_{\pm 6.84}$	$2.2482_{\pm 0.0591}$	$0.1091_{\pm 0.0524}$	$4.13_{\pm 0.05}$	$8.65_{\pm 0.09}$	$13.69_{\pm 0.14}$

From this ablation study, we conclude that each component of the DiffGoL methodology contributes a distinct and necessary role. Supervising the initial-state predictions enforces structural fidelity to the ground-truth initial states, the DiffGoL layer ensures dynamical consistency with the final states, and the Fuzziness Index penalty forces binarised predictions. It is the synergy between these three terms that yields the best overall performance.

5 Conclusions

This work presented an ablation study of the Differentiable Game of Life (DiffGoL) methodology proposed in [1] to address the reverse GoL problem. Our

Table 7. Ablation study on final states across different board sizes. Values are reported as mean with standard deviation as subscript over 15 seeds. Best and second-best results per resolution are shown in **bold** and *italics*, respectively. Reference MSEs (empty state prediction): $0.2074_{\pm 0.0905}$ (20×20) and $0.1993_{\pm 0.0862}$ (25×25).

Configuration	Fuzziness Index ↓	EMD ↓	MSE ↓
20×20 Boards			
Classic	$\mathbf{0.0680}_{\pm 0.0257}$	$4.5010_{\pm 0.0144}$	$0.2027_{\pm 0.0867}$
DiffGoL	$0.1669_{\pm 0.0850}$	$1.5695_{\pm 0.0067}$	$0.0562_{\pm 0.0348}$
Model 010	$\mathit{0.0748}_{\pm 0.0662}$	$1.6920_{\pm 3.0431}$	$\mathbf{0.0251}_{\pm 0.0556}$
Model 011	$0.1482_{\pm 0.0787}$	$\mathbf{1.3286}_{\pm 0.0220}$	$\mathit{0.0353}_{\pm 0.0240}$
Model 101	$0.0828_{\pm 0.0364}$	$3.1023_{\pm 0.0117}$	$0.1511_{\pm 0.0807}$
Model 110	$0.1841_{\pm 0.0926}$	$\mathit{1.4633}_{\pm 0.0043}$	$0.0568_{\pm 0.0366}$
25×25 Boards			
Classic	$\mathbf{0.0672}_{\pm 0.0245}$	$4.9060_{\pm 0.0206}$	$0.1952_{\pm 0.0826}$
DiffGoL	$0.1596_{\pm 0.0809}$	$1.7076_{\pm 0.0052}$	$0.0530_{\pm 0.0319}$
Model 010	$\mathit{0.0690}_{\pm 0.0594}$	$1.8935_{\pm 3.4343}$	$\mathbf{0.0234}_{\pm 0.0532}$
Model 011	$0.1412_{\pm 0.0744}$	$\mathbf{1.4312}_{\pm 0.0246}$	$\mathit{0.0326}_{\pm 0.0216}$
Model 101	$0.0798_{\pm 0.0345}$	$3.3829_{\pm 0.0167}$	$0.1418_{\pm 0.0762}$
Model 110	$0.1759_{\pm 0.0884}$	$\mathit{1.6070}_{\pm 0.0069}$	$0.0537_{\pm 0.0339}$

findings led to the following key conclusions: i) **Necessity of including the boundary conditions:** introducing periodic padding in the convolution calculations showed to improve the performance of the models, suggesting that incorporating the cellular automaton dynamics within the model is crucial. ii) **Components importance:** supervising the initial state predictions, together with the fuzziness index penalty, is crucial for achieving structural fidelity to the ground-truth initial states, while the DiffGoL layer ensures consistency with respect to the GoL dynamics. iii) **Effectiveness of the DiffGoL layer:** from the ablation study, we conclude that the combination of the three components of the DiffGoL methodology yields a strong synergy, achieving the best overall performance. The absence of any of these terms leads to failure with respect to the validation criteria.

In summary, the Differentiable Game of Life methodology provides a robust framework in which each component is crucial for achieving the best results. Future research should focus on implementing the DiffGoL layer and the fuzziness index within new architectures, as well as increasing the number of generations between the initial and final states.

Acknowledgments

The present study has been supported by the "Agencia Estatal de Investigación (España)" (grant ref.: PID2023-150663NB-C22 / AEI / 10.13039 / 501100011033). Víctor

Manuel Vargas has been supported by the Ministerio de Ciencia, Innovación y Universidades, the Agencia Estatal de Investigación and the European Social Fund Plus (grant ref.: MICIU/AEI/10.13039/501100011033, JDC2024-054787-I).

References

1. Caravaca, Á.F., Guijo-Rubio, D., Vargas, V.M.: A cnn-based approach to the reverse game of life problem. In: 11th International Work-Conference on the Interplay between Natural and Artificial Computation (IWINAC). Canary Islands, Spain (May 2026)
2. Cox, D.R.: The regression analysis of binary sequences. Journal of the Royal Statistical Society: Series B (Methodological) **20**(2), 215–232 (12 2018)
3. Gardner, M.: The fantastic combinations of john conway's new solitaire game "life". Scientific American **223**(4), 120–123 (1970)
4. Kaggle: Conway's reverse game of life. `https://www.kaggle.com/competitions/conway-s-reverse-game-of-life` (2014), online competition
5. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
6. Rubner, Y., Tomasi, C., Guibas, L.: A metric for distributions with applications to image databases. In: Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271). pp. 59–66 (1998)
7. Zaitsev, D.A.: k-neighborhood for cellular automata (2016)