

The GATTACA Framework: Graph Neural Network-Based Reinforcement Learning for Controlling Biological Networks

Andrzej Mizera^{1,2,*}, Jakub Zarzycki^{1,3}

¹Faculty of Mathematics, Informatics, and Mechanics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland

²IDEAS Research Institute, Królewska 27, 00-060 Warsaw, Poland

³IDEAS NCBR Sp. z o.o., Chmielna 69, 00-801 Warsaw, Poland

andrzej.mizera@ideas.edu.pl, jk.zarzycki@student.uw.edu.pl

Abstract

Cellular reprogramming, the artificial transformation of one cell type into another, has been attracting increasing research attention due to its therapeutic potential for complex diseases. However, identifying effective reprogramming strategies through classical wet-lab experiments is hindered by long time commitments and high costs.

Although computational methods have been proposed to address this challenge, exact state-of-the-art techniques suffer from limited scalability owing to the notorious state space explosion problem. To overcome this limitation, we explore deep reinforcement learning (DRL) for controlling holistic Boolean network models of complex biological systems, such as gene regulatory and signalling pathway networks. We formulate a novel control problem for Boolean network models operating under the asynchronous update mode, specifically tailored to the context of cellular reprogramming. To solve it, we devise GATTACA – a DRL-based computational framework explicitly designed for scalability, capable of handling large and complex network models where exact methods fail. To facilitate scalability of our framework, we consider our previously introduced concept of a pseudo-attractor and improve the procedure for effective identification of pseudo-attractor states. We incorporate graph neural networks with graph convolution operations into the artificial neural network approximator of the DRL agent’s action-value function. The new architecture allows us to leverage the available knowledge on the structure of a biological system and to indirectly, yet effectively, encode the system’s dynamics into a latent representation.

Experiments on several large-scale, real-world biological networks from the literature demonstrate the scalability and effectiveness of our approach.

Code — <https://github.com/andrzejmizera/gattaca>

Extended version — <https://arxiv.org/abs/2505.02712>

1 Introduction

Gene regulatory networks (GRNs) play a central role in determining cellular identity and function. Their emergent dynamics can be effectively modelled and comprehensively understood within the framework of Boolean networks

*Corresponding author.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

(BNs), where genes are represented as binary variables with interactions encoded as logical functions. Attractors of such networks correspond to stable gene expression profiles and are commonly interpreted as phenotypic cell states (Kauffman 1969; Huang 2002).

Disruptions in GRNs may lead cells into pathological states (Barabási, Gulbahce, and Loscalzo 2011). Cellular reprogramming seeks to restore healthy dynamics by intervening to guide the system from an undesirable attractor to a target one. However, finding effective intervention strategies through wet-lab experiments is costly and time-consuming.

Computational approaches have emerged to address this, yet exact state-of-the-art symbolic methods, like those implemented in CABEAN (Su and Pang 2021), face scalability limitations due to the exponential size of the state space. Reinforcement learning (RL), particularly deep RL (DRL), has proven powerful in high-dimensional decision problems. Previous applications of DRL to BN control have shown promise (Acerese et al. 2021; Moschoyiannis et al. 2023), yet are limited by the way the structure and dynamics of the network are encoded.

Our prior work (Mizera and Zarzycki 2025) introduced the pbn-STAC framework, which exploits DRL to control asynchronous BNs. Here, we generalise and scale this approach through GATTACA, a novel framework leveraging graph neural networks (GNNs) to embed BN structure and guide DRL control strategies. To reflect biological observability, we restrict interventions to attractor states that are identifiable in practice.

We demonstrate GATTACA’s performance on several large, real-world biological networks, showing that it identifies successful and interpretable reprogramming strategies in cases where symbolic methods fail.

2 Related Work

Control of asynchronous BNs has long relied on symbolic techniques. Exact algorithms and software tools such as CABEAN offer formal guarantees, but do not scale well due to the infamous state-space explosion. Earlier reinforcement learning efforts explored Q-learning for probabilistic BN models with limited network sizes (Sirin, Polat, and Alhajj 2013).

The idea of combining Q-learning with deep learning has recently emerged. For the first time, such a control method

was proposed for BNs in (Papagiannis and Moschoyannis 2020). It was based on the Double Deep Q Network (DDQN) with Prioritised Experience Replay (PER). Variants of this approach were later explored in (Acernese et al. 2021; Moschoyannis et al. 2023). However, most applications remain restricted to small real-world systems.

Our previous framework, pbn-STAC (Mizera and Zarzycki 2025), introduced the concept of pseudo-attractors, a procedure for identification of pseudo-attractor states, and a DRL-based control scheme for BNs and probabilistic BNs of up to 33 nodes. GATTACA builds on this foundation by incorporating GNNs into the DRL agent’s architecture, enabling improved structure-based generalisation and unprecedented scalability.

3 Preliminaries

A *Boolean Network* is defined as a set of binary nodes (genes) together with associated Boolean update rules that determine their evolution in discrete time steps. In this work, we consider the asynchronous regime, where only one node is updated at a time. A state of a BN with n nodes is a binary vector $\mathbf{x} \in \{0, 1\}^n$ specifying the activity of each node. An *attractor* is a set of states from which no escape is possible; formally, it corresponds to a bottom strongly connected component (BSCC) of the state transition graph under asynchronous updates. Attractors represent the long-term behaviours of the system and correspond to phenotypic cell states.

Since exact attractor enumeration is computationally prohibitive, we defined *pseudo-attractors* to be frequently revisited states under the stationary distribution of the induced Markov chain, as practical approximations of observable phenotypes (Mizera and Zarzycki 2025). Formally, let \mathcal{A} be an attractor of an asynchronous BN and let $\pi_{\mathcal{A}}$ be the stationary distribution of the underlying Markov chain restricted to \mathcal{A} . The pseudo-attractor associated with \mathcal{A} is the subset $P_{\mathcal{A}} \subseteq \mathcal{A}$ such that $\pi_{\mathcal{A}}(s) \geq \frac{1}{|\mathcal{A}|}$ for all $s \in P_{\mathcal{A}}$.

As shown in (Mizera and Zarzycki 2025), pseudo-attractors have the following two important properties. *Theorem 1:* For every BN attractor there exists an associated non-empty pseudo-attractor. *Theorem 2:* If $\pi_{\mathcal{A}}$ is uniform, then $P_{\mathcal{A}} = \mathcal{A}$.

We define a novel control problem tailored for cellular reprogramming, in which interventions are permitted only in pseudo-attractor states to reflect biological constraints. Given a BN, a source pseudo-attractor state, and a target configuration specifying desired values for a subset of nodes, an **attractor-target control strategy** is a sequence of instantaneous interventions that guides the network to a pseudo-attractor state aligned with the target. Interventions consists of flipping the states of selected genes and are only applied in pseudo-attractor states. A strategy of minimal length among all attractor-target control strategies is termed minimal.

Our **attractor-target control problem** is to compute a minimal attractor-target control strategy for a given source pseudo-attractor state and a target configuration. This formulation generalises the control problem introduced

in (Mizera and Zarzycki 2025), allowing transitions to any attractor that satisfies a partial specification rather than to a unique target state. It aligns with biological practice, where the goal is often to induce phenotypes characterised by partial gene expression profiles rather than fully specified states. For more details and a schematic illustration of the concept of the attractor–target control strategy, we refer the reader to the extended version of this paper.

4 The GATTACA Framework

Since the problem of BN control can be viewed as a Markov Decision Process (MDP), we decide to use a DRL agent to learn an optimal control policy mapping BN states to interventions. We propose a **Graph-based Attractor-Target Control Algorithm (GATTACA) framework** – a GNN-based DRL approach for solving the attractor-target control problem. With cellular reprogramming in mind, we focus on cellular phenotypic functional states, which are observable in experimental practice and thus allow the DRL agent to take actions only in (pseudo-)attractor states (PA states).

Since the identification of all attractor states is an NP-hard problem in itself, we improve the pseudo-attractor states identification procedure (PASIP), which was previously established in (Mizera and Zarzycki 2025). The original PASIP consists of two phases: environment pre-processing and DRL training. In the pre-processing phase, k random trajectories are simulated, each beginning with a burn-in of $n_0 = 200$ steps, followed by $n_1 = 1000$ steps during which state visitation frequencies are recorded. States visited for at least $k_1 = 5\%$ of this time are added to the pseudo-attractor list. During DRL training, additional attractors may be encountered. Fixed-point attractors are identified when the system remains in a single state for $n_2 = 1000$ steps. For multi-state attractors, a rolling buffer of the last $n_3 = 10000$ states is maintained and revisits are counted; any state revisited more than $k_2 = 15\%$ of the time is added to the pseudo-attractor list if not already recorded.

To improve upon this, our **improved PASIP (iPASIP)** incorporates entropy-based stopping criteria to detect pseudo-attractors more accurately and comprehensively. As random walks explore broadly, the entropy of the visited state distribution remains high, but as the trajectory enters an attractor basin, the state revisitation increases and the entropy decreases. This drop in entropy indicates the presence of pseudo-attractors in low-entropy regions of the state space.

Two additional checkpoints are introduced to each of the two phases of PASIP to improve the detection of large attractors that do not exhibit strong single-state revisitation. Specifically, if over 1,000,000 steps elapse without encountering a dominant state (i.e. one revisited in at least $k_1 = 5\%$ of steps), the most frequently visited state is added to the pseudo-attractor list as a representative. This modification enhances robustness in complex BN models and reduces computational overhead in lengthy simulations. The detailed outline of iPASIP is presented in the extended version of this paper.

The Attractor-Target Graph-based Controller (ATGC) constitutes the core of the GATTACA framework. ATGC

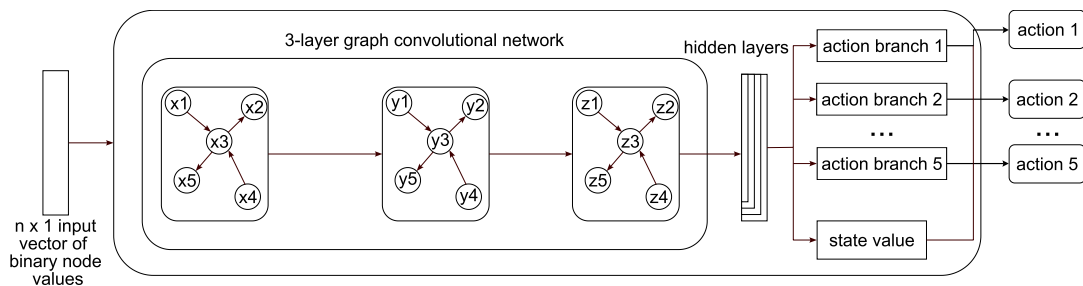


Figure 1: Schematic illustration of the ATGC DRL agent of the GATTACA framework.

consists of three main components: a Graph Convolutional Network (GCN), a multi-layer perceptron (MLP), and a Branching Dueling Q-Network (BDQ). The architecture of ATGC is schematically illustrated in Figure 1.

We propose the GCN to consist of three graph convolutional layers with ReLU activations and node-wise max-pooling to aggregate features. In our implementation, a graph convolutional layer takes the values of each input node from the previous layer and produces the corresponding output vectors for each node. The convolutional layer implements the function:

$$x'_v = \sum_{u \in N(v)} h_{\Theta}(v||u - v),$$

where v is the current node, $N(v)$ is the set of neighbours of v , and h is the graph convolutional kernel with learnable parameters Θ . In our case, h consists of a neural network with a 64-neuron deep layer and ReLU activations. Here $h(x||y)$ means conditioning-by-intervention, that is, the distribution of x with y set to some fixed value. Its role is to reduce the spurious correlations present in the graph; for more details, we refer to (Wang et al. 2019).

The reason for introducing the GCN into the architecture of the ATGC is twofold. First, to allow the DRL agent to effectively learn the interactions between the BN nodes that represent the system components. In particular, the available knowledge of the structure of the BN is explicitly incorporated into the ATGC by defining the GCN architecture in accordance with the structure graph of the BN. Then, during training, the BN is simulated in accordance with its Boolean functions. Since the resulting state of the BN is observed and used to compute the reward function discussed in the following, information on the dynamics of the BN is implicitly encoded via training into the graph convolutional kernels h_{Θ} and the remaining parameters of the ATGC architecture.

Second, the GCN creates an embedding of a discrete graph into a continuous space, which is more suitable for machine learning algorithms and eases the training of the DRL agent.

The output of the graph convolutions is passed through a three-hidden-layers MLP with 1024, 512, and 256 neurons per layer.

Finally, the DRL agent contains a Branching Dueling Q-Network (BDQ) architecture (Tavakoli, Pardo, and Kormushev 2018) to handle combinatorial action spaces, which allows multigene interventions per step. In our experiments,

we observed that the factorisation of BDQ improves sample efficiency by $\sim 30\%$ over a standard DQN. A shared GNN encoder followed by the MLP maps states to latent representations, which are fed into parallel branches estimating Q-values for each node’s flip action. This allows scalable action space representation and captures topological dependencies among genes.

The action-value function $Q(s, a)$ is optimised using double Q-learning with Huber loss and prioritised experience replay (Fujimoto, van Hoof, and Meger 2018). The reward encourages progress towards an attractor that satisfies the target configuration. It is defined as:

$$R_a(s, s') = 21 + 100 * \mathbf{1}_{TA}(s') - |a|,$$

where s and s' are the current and next PA states, respectively, $\mathbf{1}_{TA}$ is an indicator function of the target configuration, and $|a|$ is the number of genes perturbed by applying action a . This reward scheme is balanced to punish the model for perturbing many genes in a single action while providing enough reward to allow to still reach the target configuration by the means of a sequence of actions. Our experiments revealed (data not shown) that it is important that the reward function maintains a consistent sign during training (in this particular case, it is designed to remain non-negative, which is ensured by the term of 21). This design ensures convergence towards biologically plausible reprogramming strategies.

5 Experiments

We evaluate how our GATTACA framework scales with the size of the BN model. We focus on large, verified models taken from the literature for well-known real-world biological networks.

Case Studies: Models and Their Target Configurations

To evaluate the performance of our approach, we select a variety of models from the literature, ranging in size from 35 nodes (the Bladder model) to 188 nodes (the CD4⁺ model). To ensure that the target configurations used in our evaluation are both realistic and of significance to biologists, we focus on the configurations considered in the original publications where these models were introduced. The following models are considered.

Model	Size (Inputs)	iPASIP PA States			CABEAN		
		F-A	C-A	Spurious	F-A	C-A	Failed
Bladder	35 (4)	9 [20]	0 [5]	0 [0]	9 [20]	0 [4]	1 of 16
MAPK	53 (4)	6 [12]	0 [3]	0 [0]	6 [12]	0 [3]	3 of 16
T-LGL	61 (7)	149	182	2?	166	130	6 of 128
Bortezomib	67 (5)	5 [83]	0 [0]	0 [0]	5 [83]	0 [0]	0 of 32
T-diff	68 (24)	12	0	0	12	0	0 of 1
MCF-7	117 (8)	4	0	0	4	0	N/A
CD4 ⁺	188 (34)	12	0	1	12	0	0 of 1

Table 1: Comparison of the numbers of pseudo-attractor (PA) states identified by iPASIP with the numbers of attractors obtained with CABEAN. The numbers of fixpoint attractors (F-A), cyclic or complex attractors (C-A), and spurious PA states are presented for the environmental conditions enumerated in the extended version of this paper (plain numbers) and for input nodes left unspecified, i.e. all possible environmental conditions (numbers in square brackets). The numbers of environmental conditions for which CABEAN failed to compute a result are shown in the ‘Failed’ column. The ‘?’ indicates that our verification of whether the PA states are indeed spurious failed.

- A 35-node logical model for the prediction that bladder cancer cells will become invasive, originally introduced in (Remy et al. 2015) (Bladder).
- A 53-node predictive BN model of (Grieco et al. 2013) for the role of the mitogen-activated protein kinase network in urinary bladder cancer with an emphasis on the differential behaviour between EGFR overexpression and FGFR3 activating mutation (MAPK).
- A 61-node BN model of (Zhang et al. 2008) to describe the signalling involved in maintaining the long-term survival of competent cytotoxic T lymphocytes in human T cell large granular lymphocyte leukemia (T-LGL).
- A 67-node BN model constructed in (Chudasama et al. 2015) that includes major survival and apoptotic signalling pathways in U266 multiple myeloma (MM) cells to investigate the role of bortezomib, a commonly used first-line agent in MM treatment (Bortezomib).
- A 71-node comprehensive BN model originally introduced in (Naldi et al. 2010) to represent the regulatory and signalling pathways controlling the differentiation of CD4⁺ T helper (Th) cells and its reduced version of 68 nodes studied in (Su and Pang 2020) (T-diff).
- A 81-node BN model introduced in (Albert et al. 2017) to investigate the responses of the abscisic acid phytohormone signal transduction underlying stomatal closure, a vital process to reduce water loss during drought stress conditions in plants (ABA).
- A 117-node BN model of cell growth and proliferation in breast cancer incorporating common signalling pathways in the MCF-7 cell line originally introduced in (Taoma et al. 2024) (MCF-7).
- A 188-node BN model introduced in (Conroy et al. 2014) to study the role of overexpressed Caveolin-1, i.e. a vital scaffold protein heterogeneously expressed in healthy and malignant tissue, in T-cell leukaemia (CD4⁺).

The models and their corresponding target configurations are listed in Table 1 in the extended version of this paper.

Performance Evaluation Methodology

We evaluated the GATTACA framework on asynchronous BN models of various sizes to solve the attractor-target control problem. Input nodes typically represent environmental conditions and are kept fixed. Hence, they are not very interesting from the control problem point of view. For most models (i.e. Bortezomib, T-diff, MCF-7, CD4⁺), we selected biologically relevant environmental settings from the literature. For smaller models, e.g. Bladder and MAPK, a partial environmental specification was used to allow multiple conditions at once, ensuring that the control task is neither trivial nor impossible. This is due to the fact that in the majority of cases where all input nodes are set to a specific environmental condition, the models manifest a single attractor or no or all attractors aligned with the target configuration. For the T-LGL and ABA models, no environmental conditions were originally specified, so we considered all.

We let ATGC of our GATTACA framework perturb all nodes (including input nodes) except those specified in the target configuration, which we refer to as target genes. Beyond this, we do not impose any other restrictions on genes that can be perturbed: we allow full flexibility in selecting genes for interventions to ensure a fair comparison with the CABEAN benchmark and to demonstrate the potential of the GATTACA framework to identify optimal or suboptimal strategies in large networks. However, we emphasise that our framework can be straightforwardly modified to incorporate prior biological knowledge by restricting perturbations to a predefined subset of genes.

To reflect practical feasibility of wet-lab experimental procedures, we limit the number of simultaneous perturbations to five per intervention. Notice that this is a parameter of the GATTACA framework and can be adjusted to particular needs as required.

ATGC learns control strategies by minimising the number of interventions required to reach any attractor that satisfies the target configuration. Each training episode starts from a pseudo-attractor state misaligned with the target that is randomly selected from the PA states identified so far by iPASIP. Throughout each episode, we track the number

Model	Size (Inputs)	PA states	Attractors	GATTACA	CABEAN
Bladder	35 (4)	9	9	1.0	1.0
MAPK	53 (4)	6	6	1.0	1.0
T-LGL	61 (7)	333	296	1.02	1.0
Bortezomib	67 (5)	5	5	1.99	1.05
T-diff	68 (24)	12	12	2.55	1.18
ABA	81 (23)	40,000	N/A	2.23	N/A
MCF-7	117 (8)	4	4	2.33	N/A
CD4 ⁺	188 (34)	13	12	1.16	1.03
Random200	200 (0)	165	N/A	7.1	N/A

Table 2: Comparison of the average control strategy lengths required to drive the network from the source state to the target configuration obtained with the GATTACA framework and the exact ASI method implemented in CABEAN. Results that could not be obtained with CABEAN are indicated as not available (N/A).

of control actions taken, referred to as the *episode control length*, as ATGC guides the network dynamics to any PA state aligned with the given target configuration. ATGC can take up to 100 control actions per episode. If it fails to control the BN within this limit, the episode is considered unsuccessful, indicating that no valid control strategy is found.

We evaluate performance by averaging the number of steps across 10 simulations for each source PA state, as due to the nondeterministic dynamics of the environment introduced by the asynchronous update mode, results may vary between runs. The average lengths of successful control strategies are compared with the results of the state-of-the-art exact attractor-based sequential instantaneous source-target control algorithm (ASI) (Mandon et al. 2019) implemented in the CABEAN software tool (Su and Pang 2021). The ASI algorithm computes optimal strategies by leveraging symbolic techniques based on Binary Decision Diagrams (BDDs) to encode the BN dynamics.

CABEAN requires fully specified environmental conditions and searches for the shortest control paths from the source to the target attractors. However, in the case of all environmental conditions considered, e.g. the T-LGL model, we enumerate all fully specified environmental conditions and analyse them one by one with CABEAN. For each such condition, all attractors are computed and divided into source attractors, i.e. those not aligned with the target configuration, and target attractors, i.e. those aligned with the target configuration. Then, for each source attractor, the shortest path to any of the target attractors is computed with CABEAN. Finally, the averaged length of the shortest paths is reported over all environmental conditions and source attractors. Formally, the average CABEAN control path length for a BN model M , denoted $ACPL_{Cabean}(M)$, is defined as:

$$ACPL_{Cabean}(M) = \mathbb{E}_{e \in E_M} \left[\mathbb{E}_{s \in S_e^M} \left[\min_{t \in \mathcal{T}_e^M} \text{cpl}(s, t) \right] \right], \quad (1)$$

where E_M is the set of all separate environmental conditions considered for model M , S_e^M is the set of source attractors, \mathcal{T}_e^M is the set of target attractors for a given environmental condition e for model M , and cpl is the control path length computed by CABEAN for a specific pair of source and target attractors.

If there is no available target attractor for a given environmental condition, we check all other environmental conditions for a target attractor. We first switch the environmental condition and then use CABEAN to control the network within the new environmental condition. To account for the “manual” environmental condition switch, one is added to the control path length returned by CABEAN. We take the minimum length over all other environmental conditions that contain a target attractor. This allows a fair comparison with the GATTACA framework, which can handle input flipping natively.

The remaining parameters of the GATTACA framework are set as follows: Adam lr = $1e-4$, $\gamma = 0.99$, batch size = 128, replay = 10^6 , PER $\alpha = 0.6$, $\beta = 0.4 \rightarrow 1.0$, target update $\tau = 0.01$, ϵ -greedy ($1.0 \rightarrow 0.05$, 10^6 steps), and gradient clip = 10.

6 Results

We evaluate the GATTACA framework on the BN models presented. The results obtained demonstrate that our approach is scalable – it is capable of controlling environments across a wide range of sizes, particularly BN model characterised by huge state spaces.

Evaluation of iPASIP

Our iPASIP method provides a strong alternative to traditional methods for finding BN attractor states. Table 1 presents the summarised results for each model in which CABEAN was able to compute the attractors for the environmental conditions specified in Table 2 in the extended version of this paper.

To provide a more comprehensive evaluation of iPASIP, we additionally ran it on the Bladder, T-LGL, and Bortezomib models without specifying the environmental condition. For comparison, we computed the attractors for each model with CABEAN in all individual environmental conditions one by one, i.e. in 16, 128, and 32 conditions, respectively. The numbers of attractors in all environmental conditions are presented in square brackets in Table 1.

In the case of the MAPK model, CABEAN failed in 3 of 16 possible environmental conditions. Running iPASIP under all environmental conditions resulted in a very large

number of identified PA states, which we could not verify due to the lack of ground-truth information. However, running iPASIP under all environmental conditions except the ones in which CABEAN failed resulted in fully correct identification of all attractors, see Table 1.

For the Bladder model, iPASIP found 25 PA states in total in all 16 environmental conditions, while CABEAN returned 24 attractors in 15 environmental conditions and failed in one. The 24 CABEAN-found attractors are represented within the 25 PA states. By computing an attractor reachable from the extra PA state identified by iPASIP, we found that it is an attractor state of an attractor consisting of 184,320 states under the environmental condition that CABEAN was unable to handle.

We further investigated the distribution of revisits to the individual states of this large attractor. We simulated a trajectory of 1,000,000 steps with the PA state as the initial state. The numbers of revisits for individual states are shown in Figure 2. The extra PA state is clearly dominating all other attractor states in terms of the count of revisits and is correctly identified by the iPASIP method despite the attractor being exceptionally large.

In the cases of the T-LGL and CD4⁺ models, iPASIP overestimates the number of attractor states, i.e. some of the PA states are spurious attractor states, although for the T-LGL model we were unable to confirm that indeed the two states are spurious: we failed to compute attractors reachable from the potentially spurious states with a graph traversal approach. Nevertheless, as argued above, the additional states stand out due to being frequently revisited. Thus, they can still be considered viable for performing control actions.

The extra PA state¹ of the CD4⁺ BN model is indeed spurious – this can be inferred from the fact that CABEAN determined all attractor states for the given environmental condition. By simulating 1,000,000-step trajectories from this spurious attractor state, we observed that it was revisited more than 996,000 times and the trajectories consisted of less than 70 distinct states. Clearly, in this sense, the state resembles a stable state.

We verified that for all models except T-LGL all the attractors found by CABEAN have a representative amongst the PA states. For T-LGL, 143 of 166 CABEAN fixed-point attractors and 71 out of 130 CABEAN cyclic or complex attractors are detected by iPASIP, some of them multiple times as more than one PA state may belong to the same attractor (data not shown). The remaining fixed-point PA states found by iPASIP are attractor states in the six environmental conditions in which CABEAN failed².

In summary, the iPASIP approach provides a scalable and effective solution for identifying true PA states in large-size BNs. Moreover, as the case studies show, iPASIP often successfully identifies all the attractor states of a large BN.

¹In the spurious attractor state the nodes Cofilin, GATA3, IL4, PI3K, proliferation, AKT, PIP3_345, Dec2, IRF4, IL4RA, PDK1, and IKK are 1 and all the remaining nodes are 0.

²By failure we mean the fact that CABEAN ended with an error.

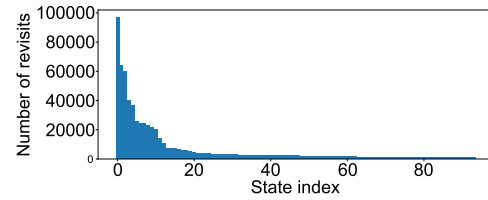


Figure 2: Bar plot of the number of revisits to individual states of the Bladder BN model attractor containing the extra PA state identified by iPASIP under the environmental condition of EGFR_stimulus set to 1 and all remaining input nodes set to 0. For readability, only states with at least 1,000 revisits are shown in the decreasing order of revisit count. State of index 0 is the PA state found by iPASIP and the remaining attractor states are indexed arbitrarily.

Evaluation of the GATTACA Framework

With regard to control problem solutions found by our GATTACA framework, the average length of the control strategies is typically within one control action/intervention of the optimal solution computed in accordance with the formula in Equation (1) based on the control path lengths obtained by the exact ASI algorithm implemented in the CABEAN software tool, as shown in Table 2.

However, for the ABA model the number of environmental conditions is too large (2^{23}) to be handled by CABEAN (entries marked N/A). As obtained with the GATTACA framework, the ABA model is characterised by a very large number of PA states. However, the GATTACA framework successfully identifies effective control strategies, demonstrating the scalability of our framework in this respect. Moreover, we conducted experiments in which the number of PA states was overestimated by a factor of 1000 and GATTACA was still able to identify effective control strategies. Those two facts prove that the GATTACA framework is resilient to both high attractor counts and spurious PA states.

To further investigate the scalability potential of the GATTACA framework, we analysed the MCF-7 model under all possible environmental conditions. It is characterised by a large number of PA states and the GATTACA framework is still capable of successfully identifying control strategies, as shown in Table 3. Unfortunately, we were unable to obtain exact results where all 256 fully specified environmental conditions were considered with CABEAN as it could not complete the attractors calculations in 168 hours (one week) on an 8-Core 2.25 GHz AMD EPYC 7742 processor with 256 GB of RAM.

Next, we analysed the original T-diff model with 71 nodes, introduced in (Naldi et al. 2010), under two environmental conditions: 1) the original, partially specified (PS) condition from (Naldi et al. 2010) and 2) the fully specified (FS) condition from (Su and Pang 2020) presented in the extended version of this paper. The former condition leaves some of the input nodes unspecified with respect to the latter.

The original 71-node T-diff model, under both conditions, is characterised by a large number of PA states identified by iPASIP. CABEAN fails to identify the attractors and can-

Model	Size	PA states	GATTACA
MCF-7	117	1,500	2.11
T-diff in PS env. cond.	71	156,864	6.17
T-diff in FS env. cond.	71	25,856	6.17

Table 3: Average control strategy lengths obtained with the GATTACA framework for the MCF-7 model under all environmental conditions and the original 71-node T-diff model of (Naldi et al. 2010) under two environmental conditions (env. cond.), i.e. the partially specified (PS) one in (Naldi et al. 2010) and the fully specified (FS) one in (Su and Pang 2020). Both conditions are characterised by large number of PA states identified by iPASIP.

not compute control strategies for these BNs. However, the GATTACA framework successfully identifies control strategies in both cases. The average control strategy length is 6.17, as shown in Table 3. It is noticeably larger than for all other BN models considered in this study. Nevertheless, this is justifiable, as such large numbers of PA states suggest that the two BN models have large numbers of attractors. Meanwhile, the size of the original T-diff model (71 nodes) is comparable to the size of the model in (Su and Pang 2020) (68 nodes), the latter with only 12 attractors; see Table 2. The state space of the original T-diff model under the two environmental conditions is then highly probable to be divided into much finer-grained basins of attraction than in the case of the modified model in (Su and Pang 2020). Consequently, driving the dynamics towards a PA state aligned with the given target configuration requires an action that places the network dynamics into a relatively small basin of attraction, likely necessitating simultaneous perturbation of many genes. Since this is restricted due to the limit imposed on the maximum number of genes that can be perturbed at once within a single action, the identified control strategies must traverse multiple attractors to reach the target configuration.

Finally, we performed an additional scalability test of the GATTACA framework on a randomly generated BN model of 200 nodes (Random200). The model is challenging as it has no input nodes and its structure graph consists of a single SCC. Therefore, the attractor detection algorithm and the ASI control algorithm in CABEAN cannot exploit the SCC-decomposition and CABEAN fails to compute the control strategies within 168 hours (one week). The GATTACA framework identifies 165 PA states and returns control strategies of average length of 7.1 for the target configuration $x_3 = 1$, where x_3 is one of the nodes; see Table 2. This relatively long control strategy on average can be explained by the fact that a network of highly interconnected nodes may require a simultaneous perturbation of a larger number of nodes than in the case of a loosely connected network. Since we limit the number of simultaneous perturbations, the GATTACA framework finds longer control strategies that traverse a number of intermediate PA states. Training of the GATTACA framework on the 200-node network required approximately 2.5h (A100) and 0.6M steps.

7 Conclusions

We formulate the attractor-target control problem in the context of cellular reprogramming – a novel control problem for biological networks modelled using the BN framework under the asynchronous update mode. To provide a scalable solution, we develop and implement the GATTACA framework. Our framework can handle models of sizes that are prohibitive for existing state-of-the-art control algorithms.

To facilitate scalability, the GATTACA framework is based on DRL. Importantly, to leverage the known structure of a biological system and to encode the network dynamics into a latent representation, we incorporate graph neural networks with graph convolution operations. The available knowledge of the structure of the a BN model is explicitly incorporated into the ATGC agent by defining the GCN architecture in accordance with the BN structure graph. Through interactions with the BN environment, the BN dynamics is implicitly encoded into the agent via training. Moreover, the GCN creates an embedding of a discrete graph into a continuous space, which enhances the training of the DRL agent.

Moreover, scalability is achieved by considering the previously introduced concept of a pseudo-attractor and by exploiting iPASIP, which we develop by improving the previously proposed approach for effective identification of PA states in large BN models. We demonstrate the effectiveness of iPASIP on a number of BN models of real-world biological networks of different sizes by comparing the identified PA states with the attractors obtained with an exact state-of-the-art algorithm implemented in the CABEAN software tool. We show that iPASIP is capable of correctly identifying the representative attractor states of all attractors in most of the cases. It also manages to detect attractor states in BN models for which CABEAN fails to compute the attractors. Furthermore, iPASIP proves its scalability and effectiveness as part of the GATTACA framework, where the determined PA states facilitate the computation of control strategies for large BN models of various real-world biological networks.

The proposed GATTACA framework offers a scalable and effective solution to the attractor-target control problem, as demonstrated by the results of computational experiments conducted on a number of large real-world biological networks. Given the source PA state and target configuration of genes, our framework finds proper control strategies that drive the network from the source PA to the target configuration by taking actions (intervening) only in intermediate PA states that correspond to phenotypical cellular states that can be observed in the lab. Our framework is capable of finding optimal or close to optimal control strategies, where in the latter case the lengths in most of the cases differ on average from the optimal strategies by at most one intervention. Moreover, the GATTACA framework is capable of identifying effective control strategies of reasonable lengths in BN models on which CABEAN fails. An additional evaluation conducted on a challenging network which structure graph is a single large SCC of 200 nodes, the analysis of which is beyond the capabilities of state-of-the-art exact solutions, further demonstrates the unprecedented scalability potential of the GATTACA framework in controlling large Boolean models of complex biological networks.

Acknowledgments

This research was funded in whole or in part by the National Science Centre, Poland under the OPUS call in the Weave programme, grant number 2023/51/I/ST6/02864.

References

- Acernese, A.; Yerudkar, A.; Glielmo, L.; and Vecchio, C. D. 2021. Reinforcement Learning Approach to Feedback Stabilization Problem of Probabilistic Boolean Control Networks. *IEEE Control Systems Letters*, 5(1): 337–342.
- Albert, R.; Acharya, B. R.; Jeon, B. W.; Zañudo, J. G. T.; Zhu, M.; Osman, K.; and Assmann, S. M. 2017. A new discrete dynamic model of ABA-induced stomatal closure predicts key feedback loops. *PLoS Biology*, 15(9): e2003451.
- Barabási, A.-L.; Gulbahce, N.; and Loscalzo, J. 2011. Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, 12(1): 56–68.
- Chudasama, V. L.; Ovacik, M. A.; Abernethy, D. R.; and Mager, D. E. 2015. Logic-Based and Cellular Pharmacodynamic Modeling of Bortezomib Responses in U266 Human Myeloma Cells. *The Journal of Pharmacology and Experimental Therapeutics*, 354(3): 448–458.
- Conroy, B. D.; Herek, T. A.; Shew, T. D.; Latner, M.; Larson, J. J.; Allen, L.; Davis, P. H.; Helikar, T.; and Cutucache, C. E. 2014. Design, Assessment, and *in vivo* Evaluation of a Computational Model Illustrating the Role of CAV1 in CD4⁺ T-lymphocytes. *Frontiers in Immunology*, 5: Article 599.
- Fujimoto, S.; van Hoof, H.; and Meger, D. 2018. Addressing Function Approximation Error in Actor-Critic Methods. arXiv:1802.09477.
- Grieco, L.; Calzone, L.; Bernard-Pierrot, I.; Radvanyi, F.; Kahn-Perlès, B.; and Thieffry, D. 2013. Integrative Modelling of the Influence of MAPK Network on Cancer Cell Fate Decision. *PLoS Computational Biology*, 9(10): e1003286.
- Huang, S. 2002. Regulation of Cellular States in Mammalian Cells from a Genomewide View. In Collado-Vides, J.; and Hofestädt, R., eds., *Gene Regulation and Metabolism: Post-Genomic Computational Approach*, 181–220. Cambridge, MA: MIT Press.
- Kauffman, S. A. 1969. Homeostasis and Differentiation in Random Genetic Control Networks. *Nature*, 224(5215): 177–178.
- Mandon, H.; Su, C.; Haar, S.; Pang, J.; and Paulevé, L. 2019. Sequential Reprogramming of Boolean Networks Made Practical. In Bortolussi, L.; and Sanguinetti, G., eds., *Proc. 17th International Conference on Computational Methods in Systems Biology (CMSB'19)*, volume 11773 of *Lecture Notes in Computer Science*, 3–19. Cham: Springer.
- Mizera, A.; and Zarzycki, J. 2025. pbn-STAC: Deep reinforcement learning-based framework for cellular reprogramming. *Theoretical Computer Science*, 1049: 115382.
- Moschogiannis, S.; Chatzaroulas, E.; Šliogeris, V.; and Wu, Y. 2023. Deep Reinforcement Learning for Stabilization of Large-Scale Probabilistic Boolean Networks. *IEEE Transactions on Control of Network Systems*, 10(3): 1412–1423.
- Naldi, A.; Carneiro, J.; Chaouiya, C.; and Thieffry, D. 2010. Diversity and Plasticity of Th Cell Types Predicted from Regulatory Network Modelling. *PLoS Computational Biology*, 6(9): e1000912.
- Papagiannis, G.; and Moschogiannis, S. 2020. Learning to Control Random Boolean Networks: A Deep Reinforcement Learning Approach. In Cherifi, H.; Gaito, S.; Mendes, J. F.; Moro, E.; and Rocha, L. M., eds., *Proc. Complex Networks and Their Applications VIII (COMPLEX NETWORKS 2019)*, volume 881 of *Studies in Computational Intelligence*, 721–734. Cham: Springer.
- Remy, E.; Rebouissou, S.; Chaouiya, C.; Zinovyev, A.; Radvanyi, F.; and Calzone, L. 2015. A Modeling Approach to Explain Mutually Exclusive and Co-Occurring Genetic Alterations in Bladder Tumorigenesis. *Cancer Research*, 75(19): 4042–4052.
- Sirin, U.; Polat, F.; and Alhajj, R. 2013. Employing Batch Reinforcement Learning to Control Gene Regulation Without Explicitly Constructing Gene Regulatory Networks. In *Proc. 23rd International Joint Conference on Artificial Intelligence*, 2042–2048. AAAI Press.
- Su, C.; and Pang, J. 2020. Sequential Temporary and Permanent Control of Boolean Networks. In Abate, A.; Petrov, T.; and Wolf, V., eds., *Proc. 18th International Conference on Computational Methods in Systems Biology*, volume 12314 of *Lecture Notes in Computer Science*, 234–251. Cham: Springer.
- Su, C.; and Pang, J. 2021. CABEAN: A software for the control of asynchronous Boolean networks. *Bioinformatics*, 37(6): 879–881.
- Taoma, K.; Ruengjitchatchawalya, M.; Liangruksa, M.; and Laomettachtit, T. 2024. Boolean modeling of breast cancer signaling pathways uncovers mechanisms of drug synergy. *PLoS ONE*, 19(2): e0298788.
- Tavakoli, A.; Pardo, F.; and Kormushev, P. 2018. Action Branching Architectures for Deep Reinforcement Learning. In McIlraith, S. A.; and Weinberger, K. Q., eds., *Proc. The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 4131–4138. AAAI Press.
- Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics*, 38(5): 1–12.
- Zhang, R.; Shah, M. V.; Yang, J.; Nyland, S. B.; Liu, X.; Yun, J. K.; Albert, R.; and Loughran, T. P. 2008. Network model of survival signaling in large granular lymphocyte leukemia. *Proceedings of the National Academy of Sciences*, 105(42): 16308–16313.