Energy Games with Weight Uncertainty*

Orna Kupferman and Naama Shamash Halevy

The Hebrew University, Israel

Abstract. An *energy game* is played between two players, modeling a resource-bounded system and its environment. The players take turns moving a token along a finite graph. Each edge of the graph is labeled by an integer, describing how its traversal affects the energy level of the system. The system wins if it never runs out of energy.
We introduce and study *energy games with weight uncertainty* (EGWUs), where the exact updates to the energy level are not a-priori known to the system. Instead, an EGWU specifies, for some subsets of edges, upper and lower bounds for their joint weight. EGWUs thus model settings in which there is only an estimation of the effect of some actions or sets of actions on the energy level, for example due to uncertainty about road conditions for an autonomous car or about the location of docking stations for a robot patrolling a warehouse. The system wins an EGWU if it has a strategy to never run out of energy, no matter what the weights are, as long as they respect the bounds. The environment wins if there are weights that respect the bounds with which it can cause the system to run out of energy.
Unlike uncertainty about the exact location of the token, which persists during the interaction, the weight of an edge is revealed upon its traversal. The fact the system learns the weights during the interaction makes EGWUs interesting, and we study the memory required to the players, determinacy of the game, and the possibility of coping with uncertainty by a larger initial energy. We give tight complexity bounds to the problems of deciding whether the system or the environment wins, and we study the effect of parameters like the richness of the function estimating the weights, or the distribution of control along the interaction.

1 Introduction

A reactive system interacts with its environment and should behave correctly in all environments. Synthesis of a reactive system thus corresponds to finding a winning strategy in a *two-player game* between the system and the environment [3]. The game is played on a graph whose vertices are partitioned between the players. Starting from some initial vertex, the players take turns moving a token along the graph. In each turn, the player that owns the current vertex decides which action to take, which amounts to choosing an edge along which the token moves to the next vertex. Together, the players generate an infinite path in the game graph.

The winning condition in the game is induced by the specification for the system. For example, using ω-regular winning conditions, one can synthesize systems that have desired on-going behaviors in all environments [18]. Adding weights to the

* This research is supported by the European Research Council, Advanced Grant ADVANSYNT.

edges of the game graph enables the winning condition to express also quantitative specifications, referring, for example, to the limit sum or average of the weights along the graph [13]. Then, a winning strategy for the system, namely a mechanism that directs it how to move the token in a way that ensures the satisfaction of the winning condition, corresponds to a system that satisfies its specification in all environments.

In *energy games* (EGs), we are interested in the ability of a *resource-bounded* system to maintain an interaction with the environment without running out of resources. Each edge of the game graph is labeled by an integer, describing its *weight* – the update to the energy level of the system that occurs whenever the edge is traversed. The game starts in some initial vertex with the system having some initial energy level. When the players move the token, the energy level of the system is updated according the weights of the edges traversed. The system wins if it never runs out of energy in the generated path. The term "energy" may refer to a wide range of applications: an actual energy level, where actions involve consumption or charging of energy; storage, where actions involve using or freeing of disc space; money, where actions involve costs and rewards to a budget of some economic entity, and more [10]. It is shown in [16] that EGs are determined, thus the system or the environment have a winning strategy, and, assuming weights given in unary, the game can be decided in polynomial time.

Different applications have led to extensions of basic EGs. For example, addressing a combination of bounded resources with behavioral requirements, researchers have studied *energy parity games*, whose winning conditions combine quantitative and qualitative conditions [7, 1]. Then, addressing systems with several bounded resources, researchers have studied *generalized energy games*, in which the system player has a multi-dimensional energy level, the updates along the edges are vectors of integers, and the system wins if it does not run out of energy in any of its resources [9]. Finally, for settings in which the environment is also resource bounded, actions in *both-bounded energy games* change the (multi-dimensional) energy level of both the system and the environment [14].

Sometimes, the system may observe only a subset of the variables that participate in the interaction. For example, in a game modeling a robot that navigates a warehouse, the robot may not be able to observe the precise location of other moving objects, such as human workers or other robots, and should still fulfill its mission while avoiding collisions. This has led to extensive studies of two-player games with *incomplete information* [15, 19]. The system wins a game with incomplete information if it has a strategy (that obviously depends only on the observable variables) such that no matter how the environment behaves and what the missing information is, the generated path satisfies the winning condition. From a technical point of view, incomplete information as above leads to uncertainty about the location of the token in the game graph. Consequently, algorithms for games with incomplete information maintain sets of indistinguishable vertices, making them (typically, exponentially) more complex than games with full information, and causing the synthesized systems to be larger [8, 2, 12].

In [11], the authors studied energy games with incomplete information. There, uncertainty makes the problem much harder: deciding the winner is Ackerman-complete [17], and determining whether there is some initial energy that is sufficient for the system to win is undecidable [11].

We introduce and study *energy games with weight uncertainty* (EGWUs), where the incomplete information concerns the updates to the energy level, rather than the location of the token. Consider again the robot that navigates an area. Factors like the type of terrain, the surface condition, and elevation gain or loss induce some estimation on the energy consumption of the robot along some routes, but the exact consumption may not be known. The estimations may refer to segments of routes or their combination, for example when we have an estimated location for a docking station or for the height difference between certain locations. Accordingly, while an EG includes a weight function $w : E \to \mathbb{Z}$ that maps each edge of the game graph to the change in the system's energy when the edge is traversed, in an EGWU we have instead a partial function $\tau : 2^E \to \mathbb{Z} \times \mathbb{Z}$ that specifies, for some subsets of edges, in particular for all singletons, upper and lower bounds on the sum of their weights. The system wins an EGWU if it has a strategy to win against all environments, and no matter which weights are used, as long as they respect the bounds. The environment wins if there are weights that respect the bounds with which it can cause the system to run out of energy. Note that since the weights are part of the setting and are only not known to the system, their choice is done offline, before the game starts.

We argue that EGWU capture realistic settings. In addition to the example above of planning in unknown environments, our work is motivated by *dynamic cloud environments*, where the cost of executing a task (e.g., in terms of CPU or memory) may only be known once the task has run, due to shared infrastructure and workload fluctuations. Then, in *energy-aware mobile apps*, apps often rely on estimated energy usage (e.g., for network or GPS access), but actual energy costs depend on background processes or current hardware state. The cost is only revealed upon execution, and systems must adapt. Note that while the cost of a certain action may not be fixed, its cost in a particular context is fixed, and corresponds to a single edge in an EGWU whose state space models the different contexts.

Unlike uncertainty about the exact location of the token, which persists during the game, the weight of an edge is revealed upon its traversal. Thus, during the interaction, the system learns the weight of traversed edges. This suggests that a system may need memory in order to win an EGWU. This is in contrast with EGs, where memoryless strategies, namely ones that only depend on the current location of the token, are sufficient for the system to win [4, 5]. We show that indeed, each EGWU G with state space S induces an EG $\mathsf{learn}(G)$ with state space $S \times \mathcal{W}_\tau$, where $\mathcal{W}_\tau$ is the set of *learned weights functions* that respects τ, namely partial weight functions that can be extended to complete ones that respect τ. The size of $\mathcal{W}_\tau$ is exponential in G, implying the system may need an exponential memory in order to with G, which we show to be tight.

While the state space of $\mathsf{learn}(G)$ is exponential in G, its diameter is only polynomial, which leads to a PSPACE algorithm for deciding whether the system can win G, which we show to be tight. By bounding the initial energy that may be required to Player 1 in order to win G, we get a PSPACE algorithm also for the problem of deciding whether such a finite initial energy exists. We also study cases in which weight uncertainty can be compensated by extra initial energy. Intuitively, the initial energy enables the systems to learn the weights, but in order for this to always succeed, the game should include no traps for the system. As for the environment, we show that memoryless strategies are sufficient for it, making the problem of deciding whether the environment wins NP-complete. This is also the complexity of deciding whether Player 2 can win the game whatever the initial energy is. We show that our lower bounds apply also for simple classes of EGWUs, for example when the system controls all the vertices of the game or when τ involves only one global estimation on the set of all edges. We also study a variant of EGWUs in which the missing weights are determined in an online manner, thus weights may be assigned in a hostile manner that depends on the history of the game so far. We show that unlike usual (offline) EGWUs, the online variant is determined.

2 Preliminaries

2.1 Energy Games

An *energy game* (EG, for short) is a two-player game $G = \langle S_1, S_2, s_0, E, b, w\rangle$, where S_1 and S_2 are disjoint sets of vertices, controlled by Player 1 and Player 2, respectively, and we let $S = S_1 \cup S_2$. Then, $s_0 \in S$ is an initial vertex, and $E \subseteq S \times S$ is a total edge relation, thus for every $s \in S$, there is $s' \in S$ such that $\langle s, s'\rangle \in E$. For $j \in \{1, 2\}$, let $E_j = E \cap (S_j \times S)$ be the set of edges that leave vertices controlled by Player j. If $S = S_j$, we say that G is *j-controlled*. The value $b \in \mathbb{N}$ is the *initial energy* of Player 1, and $w : E \to \mathbb{Z}$ is a weight function, with $w(e)$ describing the change to the energy level of Player 1 whenever an edge $e \in E$ is traversed. We extend w to sets of edges in the expected way, thus for $A \subseteq E$, we define $w(A) = \sum_{e\in A} w(e)$. Also, for an edge $\langle s_1, s_2\rangle \in E$, we sometimes abuse notations and write $w(s_1, s_2)$ rather than $w(\langle s_1, s_2\rangle)$. We define the *size* of G, denoted $|G|$, to be the size required for storing the cost function w, that is $|G| = |E| \cdot m$, where m is the largest integer in the image of w. Note that the definition corresponds to the integers in the range of w being given in unary.

In the beginning of a play in the game, a token is placed on s_0, and the energy level of Player 1 is b. Then, in each turn, the player that controls the vertex s that hosts the token chooses a successor vertex s' and moves the token along the edge $\langle s, s'\rangle$, making s' the new vertex that hosts the token. Together, the players generate a *play* $\rho = s_0, s_1, s_2, \ldots \in S^\omega$ in G, namely an infinite path that starts in s_0 and respects E: for all $i \geq 0$, we have that $\langle s_i, s_{i+1}\rangle \in E$.

Traversing an edge e updates to the energy level of Player 1 by $w(e)$. In particular, $w(e) > 0$ means that traversing e charges Player 1 with energy, and $w(e) < 0$ consumes energy. Formally, for a play ρ and an index $n \geq 0$, we define the *energy level* of ρ after n rounds, denoted $\mathcal{E}(\rho, n)$, as $b + \sum_{i=0}^{n-1} w(s_i, s_{i+1})$. Note that if an edge is traversed in ρ several times, its weight contributes to the energy level after each traversal. For a finite infix $p = s_k, s_{k+1}, \ldots, s_{k+n}$ of a play, we define the *energy change* along p by $\mathcal{E}_\Delta(p) = \sum_{i=0}^{n-1} w(s_{k+i}, s_{k+i+1})$. The goal of Player 1 is not to run out of energy; that is, keep the energy level non-negative throughout the play. The goal of Player 2 is dual, namely to make Player 1 eventually run out of energy. Formally, a play ρ is winning for Player 1 if for all $n \geq 0$, it holds that $\mathcal{E}(\rho, n) \geq 0$. Otherwise, ρ is winning for Player 2.

For $j \in \{1, 2\}$, a *strategy* for Player j directs him how to move the token in vertices he controls. The direction may depend on the history of the game so far. Thus, a strategy is a function $f_j : S^* \cdot S_j \to S$ that maps prefixes of plays that end in a vertex that is controlled by Player j to possible extensions in a way that respects E. That is, for every $\rho \in S^*$ and $s \in S_j$, we have that $\langle s, f_j(\rho \cdot s)\rangle \in E$.

A *profile* is a pair $\pi = \langle f_1, f_2 \rangle$ of strategies, one for each player. The *outcome* of a profile $\pi = \langle f_1, f_2 \rangle$ is the play obtained when the players follow their strategies in π. Formally, $\mathsf{outcome}(\pi) = s_0, s_1, s_2 \ldots \in S^\omega$ is such that for all $i \geq 0$, we have that $s_{i+1} = f_j(s_0, s_1, \ldots, s_i)$, where $j \in \{1, 2\}$ is such that $s_i \in S_j$. A strategy f_1 for Player 1 is *winning* if for every strategy f_2 for Player 2, the play $\mathsf{outcome}(f_1, f_2)$ is winning for Player 1. Likewise, a strategy f_2 for Player 2 is winning if for every strategy f_1 for Player 1, the play $\mathsf{outcome}(f_1, f_2)$ is winning for Player 2. A player wins a game if he has a winning strategy.

For an EG $G = \langle S_1, S_2, s_0, E, b, w \rangle$, the *diameter* of G is the length of the longest simple path in G. By definition, if a prefix of a play in G is longer than its diameter, then this prefix includes a cycle.

2.2 Energy Games with Weight Uncertainty

In an *energy game with weight uncertainty* (EGWU, for short), the weight function is only estimated, and the exact weights are revealed during the play. Formally, rather than a weight function, an EGWU includes a partial function $\tau : 2^E \to \mathbb{Z} \times \mathbb{Z}$ that specifies for some subsets of edges, in particular for all singletons, lower and upper bounds for the sum of their weights. We call τ an *estimated weight function*, and we say that τ is *pervasive* if $\tau(\{e\})$ is defined for all edges $e \in E$. For simplicity, for an edge $e = \langle s, s' \rangle$, we sometimes write $\tau(e)$ or $\tau(s, s')$ rather than $\tau(\{e\})$ or $\tau(\{\langle s, s' \rangle\})$, respectively. Note that possibly $\tau(e) = \langle c, c \rangle$, for some $c \in \mathbb{Z}$, in which case we say that (the energy change along) e is *known* and write $\tau(e) = c$.

Formally, an EGWU is $G = \langle S_1, S_2, s_0, E, b, \tau \rangle$, where S_1, S_2, s_0, E, and b are as in EGs, and $\tau : 2^E \to \mathbb{Z} \times \mathbb{Z}$ is a pervasive estimated weight function. Let l_G be the minimal sufficient energy to traverse a single edge in G when the weight function respects τ. Formally, $l_G = \max_{e \in E}\{-c_1 : \tau(e) = \langle c_1, c_2 \rangle\}$. Note that in order for a G to be interesting, some lower bounds c_1 need to be negative. Thus, l_G is typically positive. Then, let m_G be the largest absolute value of an integer in the image of τ. Formally, $m_G = \max_{A \subseteq E}\{|c_1|, |c_2| : \tau(A) = \langle c_1, c_2 \rangle\}$. We define the *size* of G, denoted $|G|$, as the size required for storing τ; that is $|G| = 2 \cdot k \cdot m_G$, where k is the the number of sets $A \subseteq E$ for which $\tau(A)$ is defined.

A weight function $w : E \to \mathbb{Z}$ *respects* an estimated weight function τ if for every set of edges $A \subseteq E$ for which $\tau(A) = \langle c_1, c_2 \rangle$ is defined, we have that $c_1 \leq w(A) \leq c_2$. We say that τ is satisfiable iff there is at least one weight function that respects it.

For an EGWU $G = \langle S_1, S_2, s_0, E, b, \tau \rangle$ and a weight function w, let G_w denote the EG with the structure of G and weight function w. That is, $G_w = \langle S_1, S_2, s_0, E, b, w \rangle$. One natural candidate for defining the semantics of EGWU is to say that Player 1 wins G if he has a strategy that is winning in all EGs G_w, for every weight function w that respects τ. Such a semantics, however, ignores the fact that Player 1 views his energy level, and thus learns the weight of edges traversed during the play. Below we define a semantics that models a setting with such a learning.

We start with the definitions of strategies for the players. A strategy for Player 1 in an EGWU is $f_1 : (S \times \mathbb{Z})^* \cdot S_1 \to S$ that respects E. Thus, for every $\rho \in (S \times \mathbb{Z})^*$ and $s \in S_1$, we have that $\langle s, f_j(\rho \cdot s) \rangle \in E$. Note that since Player 1 learns the weight of traversed edges, the strategy depends not only on the sequence of the vertices visited so far, but also on the weights on the edges traversed.

As for Player 2, since a strategy for Player 1 is winning iff it is winning against the most hostile weight function that respects τ, we start a play by letting Player 2 choose such a function. In addition, Player 2 moves the token in vertices he controls. Thus, a strategy for Player 2 is a pair $\langle w, f_2 \rangle$, where $w : E \to \mathbb{Z}$ is a weight function that respects τ and $f_2 : (S \times \mathbb{Z})^* \cdot S_2 \to S$ respects E. We sometimes refer to w and f_2 as the *weight* and *token* components of the strategy, respectively. Note that when τ is not satisfiable, Player 2 has no strategy, and Player 1 wins the game. We thus assume that in all EGWUs, the estimated weight function is satisfiable. Note also that since w is known to Player 2, we could have defined f_2 to have only S^* in its domain, but we preferred to have f_1 and f_2 with the same types.

In order to take into an account the learned weights, we define a *play* in an EGWU as a sequence in $(S \times \mathbb{Z})^\omega$ rather than a sequence in S^ω. For a profile $\pi = \langle f_1, \langle w, f_2 \rangle \rangle$ of strategies for the players, we define $\mathsf{outcome}(\pi)$ to be the infinite play $\rho = \langle s_0, w(s_0, s_1) \rangle, \langle s_1, w(s_1, s_2) \rangle, \langle s_2, w(s_2, s_3) \rangle \ldots \in (S \times \mathbb{Z})^\omega$ such that for all $i \geq 0$, we have that $s_{i+1} = f_j(\langle s_0, w(s_0, s_1) \rangle, \langle s_1, w(s_1, s_2) \rangle, \ldots, \langle s_{i-1}, w(s_{i-1}, s_i) \rangle, s_i)$, where $j \in \{1, 2\}$ is such that $s_i \in S_j$. For $n \geq 0$, the energy level of ρ after n rounds, denoted $\mathcal{E}(\rho, n)$ is defined as in EGs. Thus, $\mathcal{E}(\rho, n) = b + \sum_{i=0}^{n-1} w(s_i, s_{i+1})$. Likewise, for a finite infix $p = \langle s_k, c_k \rangle, \langle s_{k+1}, c_{k+1} \rangle, \ldots, \langle s_{k+n+1}, c_{k+n+1} \rangle$ of a play, we define the *energy change* along p as the sum of the weights of the edges traversed from s_k to s_{k+n}, thus $\mathcal{E}_\Delta(p) = \sum_{i=0}^{n} c_{k+i}$. Finally, winning plays and strategies for Player 1 and Player 2 are defined as in EGs.

Example 1. Consider the EGWU $G = \langle S_1, S_2, v_0, E, b, \tau \rangle$ appearing in Fig. 1. Drawing EGWUs, we describe vertices in S_1 and S_2 by circles and squares, respectively. The initial vertex v_0 is marked by an incoming edge labeled by the initial energy b. Each edge e is labeled by $\tau(e)$. Additional sets for which τ is defined are listed. In the example here, τ is defined also for the set E, with $\tau(E) = \langle -1, 0 \rangle$.

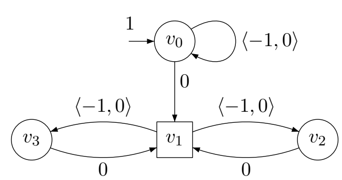

Fig. 1: The EGWU G.

Note that the three edges $\langle v_0, v_1 \rangle$, $\langle v_2, v_1 \rangle$, and $\langle v_3, v_1 \rangle$ all may have weight 0 or -1, yet $\tau(E) = \langle -1, 0 \rangle$ implies that at most one of them has weight -1, and the rest have weight 0.

We argue that Player 1 wins G. Indeed, a winning strategy f_1 of Player 1 can first take the loop on v_0. If the learned weight for it is 0, then f_1 directs Player 1 to loop in v_0 forever, keeping his energy level 1 forever. If the learned weight is -1, the energy level of Player 1 drops to 0, and f_1 directs him to move to v_1. Since the weights of both edges $\langle v_2, v_1 \rangle$ and $\langle v_3, v_1 \rangle$ must be 0, the energy level of Player 1 stays 0 forever no matter how Player 2 proceeds. □

Recall that the estimated weight function $\tau : 2^E \to \mathbb{Z}\times\mathbb{Z}$ has to be pervasive, thus specify an estimation on all sets of single edges. Note that an EGWU whose estimated weight function is defined only for single edges is not too interesting. Indeed, for such an estimated weight function there is a single *most hostile* weight function that respects it. Formally, if τ is defined only for single edges and $G = \langle S_1, S_2, s_0, E, b, \tau\rangle$, then Player 1 wins G iff Player 1 wins $G_w = \langle S_1, S_2, s_0, E, b, w\rangle$ for the weight function $w : E \to \mathbb{Z}$ in which for every edge e with $\tau(e) = \langle c_1, c_2\rangle$, we have that $w(e) = c_1$. Accordingly, such EGWUs have the same properties as EGs.

Defining τ also with respect to sets of edges enables the specification of richer estimations on the weights. As we shall show, this richness changes the theoretical and computational properties of EGs. It is interesting to analyze these properties also with respect to different classes of estimated weight functions. In particular, we study the following two special cases, where the width of τ is $|E| + 1$ and $2 \cdot |E|$, respectively. Consider an EGWU $G = \langle S_1, S_2, s_0, E, b, \tau\rangle$.

- G is *global* if τ is defined only for singletons and the set E.
- G is *flat* if each edge appears in at most two sets for which τ is defined (that is, in one set in addition to the singleton for it). Note that a global estimated weight function is flat.

For an estimated weight function τ, let $\mathcal{W}_\tau$ denote the set of functions $g : E \to \mathbb{Z}\cup\{\perp\}$ that can be extended to a function that respects τ. One can view g as a partial function from E to $\mathbb{Z}$, with $g(e) = \perp$ indicating that $g(e)$ is undefined. Intuitively, in the beginning of a play in an EGWU, Player 1 has no knowledge about the weights of the edges beyond the estimated weights induced from τ; thus Player 1 starts the game with a *learned function* g_0 such that $g_0(e) = \perp$ for all $e \in E$. Then, when the current learned function is g and the token traverses an edge e, it may be the case that the play has already used the edge e, in which case $g(e) \in \mathbb{Z}$, or this is the first traversal of e, in which case Player 1 can update his learned function to one that takes the energy change caused by the traversal. Formally, for $g \in \mathcal{W}_\tau$, $e \in E$ and $c \in \mathbb{Z}$, we define $\mathsf{update}(g, e, c)$ to be the function g' such that:

- If $g(e) \neq \perp$ then $g' = g$.
- Otherwise, g' agrees with g on all edges but e, which now has weight c. That is, $g'(e) = c$ and $g'(e') = g(e')$ for all $e' \neq e$.

Let $k = \max_{e\in E}\{|c_2 - c_1| : \tau(e) = \langle c_1, c_2\rangle\}$ be the largest uncertainty that τ imposes for a single edge. Note that since τ is pervasive, k is well defined. Also, as all the functions in $\mathcal{W}_\tau$ can be extended to functions that respect τ, we have that $|\mathcal{W}_\tau| \leq (k + 1)^{|E|}$. Thus, $\mathcal{W}_\tau$ is of size at most exponential in $|G|$.

For two learned functions $g, g' \in \mathcal{W}_\tau$, we say that g' *extends* g, denoted $g \preceq g'$, if g' is obtained from g by assigning values in $\mathbb{Z}$ to edges that are assigned $\perp$ in g. Formally, $g \preceq g'$ iff for all edges $e \in E$, if $g'(e) = \perp$ then $g(e) = \perp$, and if $g(e) \in \mathbb{Z}$ then $g'(e) = g(e)$. Clearly, the relation $\preceq$ is reflexive and transitive. Also, for every $g \in \mathcal{W}_\tau$, edge $e \in E$, and value $c \in \mathbb{Z}$ for which $\mathsf{update}(g, e, c)$ is in $\mathcal{W}_\tau$, we have that $g \preceq \mathsf{update}(g, e, z)$.

3 Theoretical Properties of EGWUs

In this section we study the theoretical properties of EGWUs. We first examine the effect of learning the weights during the interaction, and show that it can be captured by augmenting the vertices with learned weight functions. We then study the requirement to fix the weight function before the interaction starts. We show that it makes EGWUs undetermined, and that allowing Player 2 to fix the weights in an online manner makes the game determined.

3.1 From EGWUs to EGs

For an EGWU $G = \langle S_1, S_2, s_0, E, b, \tau \rangle$, we define the *EG induced from G*, denoted $\mathsf{learn}(G)$, as the EG induced from G when Player 1 maintains the weights learned during the interaction. As we shall see in Lemma 1, Player 1 wins the EGWU G iff he wins the EG $\mathsf{learn}(G)$. An analysis of $\mathsf{learn}(G)$ then provides helpful properties on the strategies required for Player 1 to win G.

Formally, $\mathsf{learn}(G) = \langle V_1, V_2, v_0, E', b, w \rangle$, where

- $V_1 = S_1 \times \mathcal{W}_\tau$. Thus, Player 1 controls vertices that correspond to vertices in G, augmented by learned weight functions.
- $V_2 = (S_2 \cup E_1) \times \mathcal{W}_\tau$. Thus, Player 2 controls two types of vertices. Vertices in $S_2 \times \mathcal{W}_\tau$, which correspond to vertices in G augmented by learned weight functions, and vertices in $E_1 \times \mathcal{W}_\tau$, which serve as intermediate vertices that enable Player 2 to decide the weight of edges in G that leave vertices in S_1 (recall that $E_1 = E \cap (S_1 \times S)$.
  Let $V = V_1 \cup V_2$. For a vertex in V of the form $(s, g) \in S \times \mathcal{W}_\tau$, we refer to s and g as the S-component and the $\mathcal{W}_\tau$-component of (s, g).
- $v_0 = (s_0, g_0)$.
- The set E' of edges is induced by these of G, and their weight is induced from the way Player 2 chooses to update the learned weight function. Formally, we define E' and the weight function $w : E' \to \mathbb{Z}$ as follows.
  - For every vertex $(s, g) \in V_1$ and edge $\langle s, s' \rangle \in E$, we add to E' an edge with weight 0 from (s, g) to $(\langle s, s' \rangle, g)$. The edge corresponds to Player 1 choosing the successor s' of s when the token is in s and the learned weights so far agree with g.
  - For every vertex $(\langle s, s' \rangle, g) \in V_2$ and for every $c \in \mathbb{Z}$ for which the function $g' = \mathsf{update}(g, \langle s, s' \rangle, c)$ is in $\mathcal{W}_\tau$, we add to E' an edge with weight $g'(s, s')$ from $(\langle s, s' \rangle, g)$ to (s', g'). The edge corresponds to Player 2 choosing c as the weight of the edge $\langle s, s' \rangle$. Note that if Player 2 has already chosen this weight, then $(\langle s, s' \rangle, g)$ has a single succesor, which is (s', g). Indeed, by the definition of $\mathsf{update}(g, \langle s, s' \rangle, c)$, if $g(s, s') \neq \bot$, then for all $c \in \mathbb{Z}$, we have that $\mathsf{update}(g, \langle s, s' \rangle, c) = g$.
  - For every vertex $(s, g) \in V_2$, edge $\langle s, s' \rangle \in E$, and $c \in \mathbb{Z}$ for which the function $g' = \mathsf{update}(g, \langle s, s' \rangle, c)$ is in $\mathcal{W}_\tau$, we add to E' an edge with weight $g'(s, s')$ from (s, g) to (s', g'). The edge corresponds to Player 2 choosing both the successor s' of s and the weight of the edge $\langle s, s' \rangle$. As in the case above, if $g(s, s') \neq \bot$, then $g' = g$, in which case the only successor of (s, g) with S-component s' is (s', g).

We now relate G and $\mathsf{learn}(G)$. A *history* in G is a prefix $h \in (S \times \mathbb{Z})^* \cdot S$ of a play in G with the last pair contributing only the vertex element. A *history* in $\mathsf{learn}(G)$ is a prefix $h' \in V^* \cdot (S \times \mathcal{W}_\tau)$ of a play in $\mathsf{learn}(G)$ that ends in a vertex in $(S \times \mathcal{W}_\tau)$.

For a history $h = \langle s_0, c_0\rangle, \langle s_1, c_1\rangle, \ldots, \langle s_k, c_k\rangle, s_{k+1}$ in G, let $g_h \in \mathcal{W}_\tau$ be the learned weight function that corresponds to h. That is, g_h is defined for exactly all edges that are traversed in h, and it maps these edges to the weights revealed during h. Formally, for all $0 \le i \le k$, we have that $g_h(s_i, s_{i+1}) = c_i$. Since h is a prefix of a play, it is guaranteed that an edge that is traversed several times is assigned the same weight in all its traversals, thus g_h is well defined.

We define a relation $\approx$ between histories in G and $\mathsf{learn}(G)$. The definition of $\approx$ proceeds by induction on the length of the histories. First, $s_0 \approx \langle s_0, g_0\rangle$. An invariant that is maintained during the definition is that $h \cdot s \approx h' \cdot \langle s', g\rangle$ implies $s = s'$, $g_{h\cdot s} = g$, and $\mathcal{E}_\Delta(h) = \mathcal{E}_\Delta(h')$. Note that the invariant is satisfied in the induction base, where $h = h' = \epsilon$. Now, for the induction step, consider histories $h \cdot s$ and $h' \cdot \langle s, g\rangle$ such that $h \cdot s \approx h' \cdot \langle s, g\rangle$. If $s \in S_1$, then for every successor s' of s and value $c \in \mathbb{Z}$ such that $\mathsf{update}(g, \langle s, s'\rangle, c)$ is in $\mathcal{W}_\tau$, we have that $h \cdot \langle s, c\rangle \cdot s' \approx h' \cdot \langle s, g\rangle \cdot \langle (s, s'), g\rangle \cdot \langle s', \mathsf{update}(g, \langle s, s'\rangle, c)\rangle$. If $s \in S_2$, then for every successor s' of s and value $c \in \mathbb{Z}$ such that $\mathsf{update}(g, \langle s, s'\rangle, c)$ is in $\mathcal{W}_\tau$, we have that $h \cdot \langle s, c\rangle \cdot s' \approx h' \cdot \langle s, g\rangle \cdot \langle s', \mathsf{update}(g, \langle s, s'\rangle, c)\rangle$. Note that when $s \in S_1$, the history in $\mathsf{learn}(G)$ is extended by two vertices, corresponding to first Player 1 choosing s', and then Player 2 choosing c. When $s \in S_2$, Player 2 chooses both s' and c, and h' is extended with one vertex that includes both choices. Note also that in both cases, the invariant about the last vertex, the learned function, and the energy update is maintained. In particular, by the induction hypothesis, we have that $\mathcal{E}_\Delta(h) = \mathcal{E}_\Delta(h')$, and so $\mathcal{E}_\Delta(h\cdot\langle s, c\rangle\cdot s') = \mathcal{E}_\Delta(h)+c = \mathcal{E}_\Delta(h')+c = \mathcal{E}_\Delta(h'\cdot\langle s, g\rangle\cdot\langle s', \mathsf{update}(g, \langle s, s'\rangle, c)\rangle)$, in the case $s \in S_1$, and similarly for the case $s \in S_2$. For plays $\rho \in (S \times \mathbb{Z})^\omega$ in G and $\rho' \in V^\omega$ in $\mathsf{learn}(G)$, we say that $\rho \approx \rho'$ if for every finite history h in ρ, there is a finite history h' in ρ' such that $h \approx h'$. By the invariant about the energy update, if $\rho \approx \rho'$, then ρ is winning for Player 1 iff ρ' is winning for Player 1.

Finally, for a play $\rho' \in V^\omega$ in $\mathsf{learn}(G)$ and a weight function $w : E \to \mathbb{Z}$, we say that w *agrees with* ρ' if w respects τ and all the learned weight functions in ρ' agree with w. Note that since all the learned weight functions are in $\mathcal{W}_\tau$ and are related by $\preceq$, such a weight function w exists.

Lemma 1. *Consider an EGWU G and the EG $\mathsf{learn}(G)$. Player 1 wins G iff he wins $\mathsf{learn}(G)$.*

Proof. Assume first that Player 1 wins G. Let $f_1 : (S \times \mathbb{Z})^* \cdot S_1 \to S$ be a winning strategy for Player 1 in G. Consider the strategy $f_1' : V^* \cdot V_1 \to V$ such that for every history $h' \cdot \langle s, g\rangle \in V^* \cdot V_1$, we have that $f_1'(h', \langle s, g\rangle) = \langle (s, f_1(h \cdot s)), g\rangle$, where h is the single history in G such that $h \approx h'$. That is, f_1' chooses the successor that f_1 chooses in the corresponding history. We prove that f_1' is winning for Player 1 in $\mathsf{learn}(G)$. Consider a strategy f_2' for Player 2 in $\mathsf{learn}(G)$, and assume by way of contradiction that $\rho' = \mathsf{outcome}(f_1', f_2')$ is not winning for Player 1. Let $w : E \to \mathbb{Z}$ be a weight function that agrees with ρ'. Let $\rho = \langle s_0, c_0\rangle, \langle s_1, c_1\rangle, \ldots \in (S \times \mathbb{Z})^\omega$ be the play in G such that $\rho \approx \rho'$. Note that such a unique play ρ exists, obtained by following the vertices in S that are visited along ρ' and their weights in w. Consider now the strategy $\langle w, f_2\rangle$ for Player 2 in G such that $f_2 : (S \times \mathbb{Z})^* \cdot S_2 \to S$ is defined, for every history $h = \langle s_0, c_0\rangle, \langle s_1, c_1\rangle, \ldots, \langle s_{k-1}, c_{k-1}\rangle, s_k \in (S \times \mathbb{Z})^* \cdot S_2$ that is a prefix of ρ, by $f_2(h) = s_{k+1}$. Then, $\mathsf{outcome}(f_1, \langle w, f_2\rangle) = \rho$. Since ρ' is not winning for Player 1 and $\rho \approx \rho'$, then ρ is not winning for Player 1 either, contradicting the fact that f_1 is winning.

Assume now that Player 1 wins $\mathsf{learn}(G)$. Let $f_1' : V_1 \to V$ be a memoryless winning strategy for Player 1 in $\mathsf{learn}(G)$. Since $\mathsf{learn}(G)$ is an EG, such a strategy exists [5, 4]. Consider the strategy $f_1 : (S \times \mathbb{Z})^* \cdot S_1 \to S$ such that for every history $h \cdot s \in (S \times \mathbb{Z})^* \cdot S_1$, we have that $f_1(h \cdot s) = s'$, where s' is such that $f_1'(\langle s, g_{h \cdot s}\rangle) = \langle (s, s'), g_{h \cdot s}\rangle$. That is, f_1 chooses the successor that f_1' chooses when s is paired with the learned weight function that corresponds to $h \cdot s$.

We now prove that f_1 is winning for Player 1 in G. Consider a strategy $\langle w, f_2\rangle$ for Player 2 in G, and assume by way of contradiction that $\rho = \mathsf{outcome}(f_1, \langle w, f_2\rangle) \in (S \times \mathbb{Z})^\omega$ is not winning for Player 1. Let $\rho' = \langle x_0, g_0\rangle, \langle x_1, g_1\rangle, \ldots \in V^\omega$ be the play in $\mathsf{learn}(G)$ such that $\rho \approx \rho'$. Note that such a unique play ρ' exists, obtained by following the vertices and the weights of the traversed edges along ρ. Consider now the strategy $f_2' : V^* \cdot V_2 \to V$ for Player 2 in $\mathsf{learn}(G)$ that is defined as follows. Consider a history $h' \cdot \langle s, g\rangle \in V^* \cdot (S \times \mathcal{W}_\tau)$ in $\mathsf{learn}(G)$ such that $h' \cdot \langle s, g\rangle$ is a prefix of ρ'. Let $h' \cdot \langle s, g\rangle = \langle x_0, g_0\rangle, \langle x_1, g_1\rangle, \ldots, \langle x_k, g_k\rangle$, and let $h \cdot s \in (S \times \mathbb{Z})^* \cdot S$ be the history in G such that $h \approx h'$. Here too, h exists and is unique. If $s \in S_2$, we define $f_2'(h \cdot \langle s, g\rangle) = \langle x_{k+1}, g_{k+1}\rangle$. If $s \in S_1$, then it must be that $x_{k+1} = \langle s, s'\rangle$ for some successor s' of s, and $g_{k+1} = g$. Then, we define $f_2'(h \cdot \langle s, g\rangle \cdot \langle x_{k+1}, g_{k+1}\rangle) = \langle s', g_{k+2}\rangle$. Note that the first case handles all the prefixes of ρ' in which Player 2 applies his strategy in vertices in $S_2 \times \mathcal{W}_\tau$, and the second case handles all the prefixes of ρ' in which Player 2 applies his strategy in vertices in $E_1 \times \mathcal{W}_\tau$. By the definition of f_2', we have that $\mathsf{outcome}(f_1', f_2') = \rho'$. Since ρ is not winning for Player 1 and $\rho \approx \rho'$, then ρ' is not winning for Player 1 either, contradicting the fact that f_1' is winning. □

3.2 Determinacy: online and offline weight functions

A two-player game is *determined* if for every instance G of the game, either Player 1 or Player 2 wins in G. Since energy objectives are closed sets, EGs are determined [16]. We show that this is not the case for EGWUs, and argue that undeterminacy is caused by the fact Player 2 chooses the weights in advance, rather than in an online manner. Formally, we show that a variant of EGWUs in which the weights are chosen online is determined.

Theorem 1. *EGWUs need not be determined. Undeterminacy holds already for* 1-*controlled and global EGWUs.*

Proof. Consider the EGWU $G_{undet} = \langle S_1, S_2, s_0, E, b, \tau\rangle$ described in Fig. 2. In addition to the estimated weights labeling the edges in the figure, we have that $\tau(E) = \langle 0, 0\rangle$. Note that since $S_2 = \emptyset$, a strategy for *Player* 2 consists only of a weight function that respects τ.

We show that neither Player 1 nor Player 2 wins G_{undet}. We start with Player 1. Consider a strategy f_1 for Player 1. If $f_1(v_0) = v_1$, then the weight function w with $w(\langle v_1, v_1\rangle) = -1$ and $w(\langle v_2, v_2\rangle) = 1$ respects τ, and the outcome of f_1 and w is winning for Player 2, as the loop on v_1 is negative. Thus, in this case, f_1 is not winning for Player 1. Similarly, if $f_1(v_0) = v_2$, then the weight function w with $w(\langle v_1, v_1\rangle) = 1$ and $w(\langle v_2, v_2\rangle) = -1$ respects τ, and results in a play that is winning for Player 2. Hence, Player 1 does not have a winning strategy in G_{undet}.

We continue to Player 2, and consider a weight function w that respects τ. Since $\tau(E) = \langle 0, 0\rangle$, it must be that $w(E) = 0$, and hence there is $i \in \{1, 2\}$ for which $w(\langle v_i, v_i\rangle) \geq 0$. Then, the strategy f_1 with $f_1(v_0) = v_i$ is such that the outcome of

f_1 and w is winning for Player 1, as the loop on v_i is non-negative. Thus, Player 2 does not have a winning strategy in G_{undet} either. □

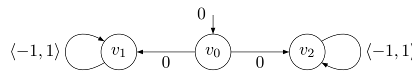

Fig. 2: The EGWU G_{undet}.

Remark 1. **[Online EGWUs]**. In an *online EGWU*, we let Player 2 choose, for each edge e, the weight $w(e)$ in the first traversal of e. Since a strategy of Player 1 is winning if it wins against all strategies of Player 2 (in particular these that happen to agree with some online strategy), then from the point of view of Player 1, usual (offline) and online EGWUs coincide.

As for Player 2, the ability to choose the weights in an online manner is helpful. For example, Player 2 has a winning strategy in the online variant of the EGWU G_{undet} described in Theorem 1. Indeed, after Player 1 chooses a loop, Player 2 can assign it a weight of -1. Moreover, the reasoning in the proof of Lemma 1 implies that Player 2 wins an online EGWU G iff he wins the EG $\mathsf{learn}(G)$. Thus, as EGs are determined, so are online EGWUs. □

4 Using Initial Energy for Learning

In this section we show that in some cases, incomplete information can be compensated by extra initial energy. Intuitively, the initial energy enables Player 1 to learn the weights, but in order for this to always succeed, the game should include no traps for Player 1.

Consider an EGWU $G = \langle S_1, S_2, s_0, E, b, \tau \rangle$. We say that Player 1 can *reset* G if, ignoring energy constraints, Player 1 has a strategy to reach s_0 from all vertices $s \in S$. We say that Player 1*wins G with complete information* if for every weight function w that respects τ, Player 1 wins the EG G_w. Clearly, if Player 1 wins G, then Player 1 also wins G with full information. The other direction, however, is not true. For example, the undetermined EGWU G_{undet} presented in the proof of Theorem 1 is such that Player 1 wins G_{undet} with complete information, yet does not win G_{undet}.

It is not surprising that an extra initial energy enables Player 1 to learn the weight function in an EGWU he can reset. Below we analyze the exact extra energy needed and a strategy for Player 1 to use it.

Recall that l_G is the minimal sufficient energy to traverse a single edge in G when the weight function respects τ. If $l_G \leq 0$, then all edges have non-negative weights, and Player 1 wins G with an arbitrary strategy and with no extra initial energy. Otherwise, we have the following.

Theorem 2. *Consider an EGWU $G = \langle S_1, S_2, s_0, E, b, \tau \rangle$. If Player 1 can reset G and wins G with complete information, then there is $\Delta_G \leq 2 \cdot l_G \cdot |E|^2$ such that Player 1 wins in G with initial energy $b + \Delta_G$. The bound on Δ_G is tight, even for 1-controlled and global EGWUs.*

Proof. Assume that Player 1 can reset G and wins G with complete information. Let $w_1, w_2, \ldots, w_n$ be an order on the weight functions that respect τ. Since Player 1 wins G with complete information, then for every $1 \le i \le n$, there is a winning strategy f'_i for Player 1 in G_{w_i}. Since EGs are memoryless, we assume that f'_i is memoryless. Also, as Player 1 can reset G, then for every vertex $s \in S$, there is a memoryless strategy f_s for Player 1 to reach s_0. Note that since f_s is memoryless, s_0 is reached along a simple path (indeed, if Player 2 can force a cycle, he can prevent Player 1 from reaching s_0).

A winning strategy f_1 for Player 1 in G with initial energy $b + 2 \cdot l_G \cdot |E|^2$ then proceeds as follows: Let $i = 1$. $(*)$ Apply f'_i from s_0. If at some point an edge $\langle s, s' \rangle$ is traversed and its weight is not $w_i(s, s')$, then apply from s' the strategy $f_{s'}$ to reach s_0, increase i to the minimal index $j > i$ such that w_j agrees with the weights revealed so far, and return to $(*)$.

We prove that f_1 is winning. For $1 \le i \le n$, we say that iteration i is *good* if during this iteration, the weights revealed for all edges traversed agrees with the weight function w_i. Note that if there is a good iteration i, then b is a sufficient energy for Player 1 in this iteration. Indeed, f'_i is winning in G_{w_i} with energy b. We claim that for every strategy f_2 for Player 2, a good iteration i exists, and that when Player 1 reaches s_0 at the beginning of this iteration, the energy level is at least b.

Let $1 \le k \le n$ be such that the weight strategy of Player 2 is w_k. Then, the iteration in which Player 1 follows w_k is good, and the strategy f_1 is guaranteed to eventually reach an iteration in which all the traversed edges agree with w_k. Also, as Player 1 returns to s_0 and starts a new iteration only after the weight of a new edge is revealed, a good iteration is guaranteed to arrive after at most $m < |E|$ iterations.

We claim that each iteration $1 \le i \le m$ that is not good reduces the energy of Player 1 by at most $2 \cdot l_G \cdot |E|$. Indeed, in each such iteration, an edge whose weight does not agree with w_k is revealed along a path p, and then the token returns to s_0 along a simple path. Notice that every cycle C in p (if there is such) has $\mathcal{E}_\Delta(C) \ge 0$, as otherwise f'_i is not winning in G_{w_i}. Let $m \in \mathbb{N}$ be such that $\mathcal{E}(p, m)$ is minimal. Thus, energy of $-\mathcal{E}(p, m)$ is sufficient for Player 1 in p. Let p' be the prefix of p up to the m-th position. Thus, $\mathcal{E}_\Delta(p') = \mathcal{E}(p, m)$. Now, let p'' be the simple path achieved from p' by removing all cycles. Since all cycles are non-negatives, $\mathcal{E}_\Delta(p'') \le \mathcal{E}_\Delta(p') = \mathcal{E}(p, m)$. Also, since p'' is simple, we have that $\mathcal{E}_\Delta(p'') \ge -l_G \cdot |E|$. Thus, $\mathcal{E}(p, m) \ge -l_G \cdot |E|$.

Thus, energy of $l_G \cdot |E|$ is sufficient for traversing p. Also returning the token to s_0 along a simple path reduces the energy level of Player 1 by at most $l_G \cdot |E|$, and so we are done.

While the bound on Δ_G can be tightened for specific EGWUs, taking into account the bounds that τ defines for sets of edges and the structure of the game graph, in some cases an extra energy of $O(l_G \cdot |E|^2)$ is needed. Below we present a family of EGWUs that attains the bound. Given $n, l \ge 1$, let $c = \sum_{i=1}^{n} i$, and consider the EGWU $G_{n,l}$ described in Fig. 3. In addition to the definition of τ for single edges in the figure, we have that $\tau(E) = (l \cdot (c - n)) - (n - 1)$. Note that $l_{G_{n,l}} = l$ and that G is 1-controlled and global.

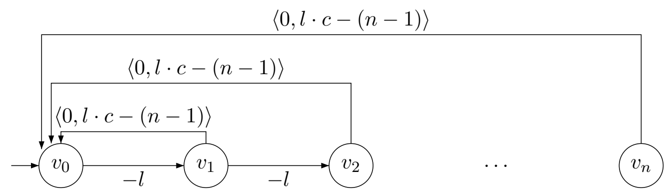

Fig. 3: The EGWU $G_{n,l}$.

We first claim that for every weight function w that respects τ, there is $1 \leq i \leq n$ such that $w(v_i, v_0) \geq l \cdot i$. Indeed, if $w(v_i, v_0) < l \cdot i$ for all $1 \leq i \leq n$, then $w(v_i, v_0) \leq (l \cdot i) - 1$, in which case $\sum_{i=1}^{n} w(v_i, v_0) \leq \sum_{i=1}^{n}((l \cdot i) - 1) = (l \cdot c) - n$. Since $w(E) = (\sum_{i=1}^{n} w(v_i, v_0)) + (\sum_{i=1}^{n} w(v_{i-1}, v_i))$, and the second sum is known to be $-(l \cdot n)$, it follows that $w(E) \leq (l \cdot (c - n)) - n$, contradicting the fact that w respects τ. The existence of $1 \leq i \leq n$ such that $w(v_i, v_0) \geq l \cdot i$ implies that Player 1 wins G with complete information with the initial energy of $l \cdot n$. Indeed, since $w(v_i, v_0) \geq l \cdot i$, then $\mathcal{E}_\Delta(\langle v_0, -l\rangle, \langle v_1, -l\rangle, \ldots, \langle v_{i-1}, -l\rangle, \langle v_i, w(v_i, v_0)\rangle, v_0) \geq 0$, and so Player 1 has sufficient initial energy to reach v_i and then loop forever in the cycle that goes back to v_0 via v_i.

On the other hand, we argue that when Player 1 has uncertainty about the weights, then for every strategy f_1 of Player 1, there is a weight strategy w_{f_1} for Player 2 such that Player 1 runs out of energy in $\mathsf{outcome}(f_1, w_{f_1})$ unless his initial energy is at least $l \cdot c$. Since the initial energy of Player 1 in G is $l \cdot n$, we get that $\Delta_G = l \cdot (c - n)$, which is $O(l \cdot n^2)$.

Consider a strategy f_1 for Player 1. We define w_{f_1} as follows. Consider first the weight strategy w such that $w(v_i, v_0) = 0$ for all $1 \leq i \leq n$. Note that w does not respect τ. Let $i_1, i_2, .., i_{n-1}$ be the order in which f_1 traverses the first $n - 1$ edges leading to v_0 when edges are weighted by w. That is, for all $1 \leq j_1 < j_2 \leq n - 1$, the first traversal of the edge $\langle v_{i_{j_1}}, v_0\rangle$ in $\mathsf{outcome}(f_1, w)$ is prior to the first traversal of the edge $\langle v_{i_{j_2}}, v_0\rangle$. Let j be such that the edge $\langle v_j, v_0\rangle$ is the last one to be traversed in $\mathsf{outcome}(f_1, w)$. Thus, $\{j\} = [n] \setminus \{i_1, i_2, .., i_{n-1}\}$. We define $w_{f_1}(v_j, v_0) = l \cdot c - (n-1) = \tau(E) + l \cdot n$, $w_{f_1}(v_i, v_0) = 0$ for all $1 \leq i \neq j \leq n$, and $w_{f_1}(v_i, v_{i+1}) = -l$, for all $1 \leq i \leq n - 1$.

Note that w_{f_1} respects τ. Also, when Player 1 traverses the edge $\langle v_j, v_0\rangle$ for the first time, he has already traversed all the edges $\langle v_i, v_0\rangle$ for $i \neq j$. Each such traversal requires a traversal of a cycle with i edges with weight $-l$ (leading from v_0 to v_i), and one edge from v_i to v_0 with weight 0. Thus, the prefix p of the play traversed has $\mathcal{E}_\Delta(p) \leq -l \cdot c$, and so, the required initial energy for Player 1 is at least $l \cdot c$. □

The key to the learning performed in the proof of Theorem 2 is the ability of Player 1 to avoid cycles that may be negative. Below we formalize this intuition further, and show that it is tight. Consider an EGWU $G = \langle S_1, S_2, s_0, E, b, \tau\rangle$. We say that G is *permissive* if $\tau(A) \geq 0$, for every $A \subseteq E$ such that there is path in G that visits exactly all edges in A, and these are the only sets for which τ is defined. In particular, τ is not pervasive. Thus, as long as Player 1 avoids traversing the same edge twice, his energy level is guaranteed to be non-negative. Indeed, when G is permissive, then every simple path p in G that starts in s_0 satisfies $\mathcal{E}_\Delta(p) \geq 0$: In simple paths every edge is taken at most once. Then, if $p = s_0, s_1, \dots s_n$ is a simple path, under every weight function w that respects τ, we have that $\mathcal{E}_\Delta(p) = w(A)$ where $A = \{\langle s_i, s_{i+1}\rangle : 0 \leq i \leq n-1\}$. Since A is a set of edges that contains exactly all edges in the simple path p, we have that $w(A) \geq 0$. As we state below, not only the lack of traps enables Player 1 to maintain his energy non-negative G, traps enable Player 2 to win.

Theorem 3. *Permissive EGWUs are determined. Player* 1 *wins a permissive EGWU iff he has a strategy to visit its initial vertex infinitely often.*

Proof. Consider an EGWU $G = \langle S_1, S_2, s_0, E, b, \tau\rangle$. Assume first that Player 1 has a strategy f_1 to visit s_0 infinitely often. By [21], we can assume that f_1 is memoryless. We claim that f_1 is winning in G. Indeed, for every strategy $\langle w, f_2\rangle$ of Player 2, we have that $\rho = \mathsf{outcome}(f_1, \langle w, f_2\rangle)$ is a concatenation of simple cycles from s_0 back to s_0. For every such simple cycle and every prefix p of it, we have $\mathcal{E}_\Delta(p) \geq 0$. Hence, for all $n \geq 1$, we have $\mathcal{E}(\rho, n) \geq 0$, and we are done.

For the second direction, assume that Player 1 does not have a strategy to visit s_0 infinitely often. Since repeated-reachability games are determined, it follows that Player 2 has a strategy f_2 to force s_0 to be visited only finitely often. Let $E' \subseteq E$ be the set of edges going out from s_0. Consider the cost function w that assigns weight $|E|$ to each edge in E' and assigns -1 to each edge in $E \setminus E'$. We first argue that w respects τ. Indeed, for every path p from s_0 in $\mathcal{G}$, we have that $w(p) = \sum_{e\in p} w(e) = \sum_{e \in p\cap E'} w(e) + \sum_{e\in p\cap(E\setminus E')} w(e) = |p \cap E'| \cdot |E| - |\pi \cap (E \setminus E')| \geq |E| - |E| = 0$. Note that the last inequality holds since the first edge in p is in E', and the size of $E \setminus E'$ is at most E.

We prove that $\langle w, f_2\rangle$ is winning for Player 2. Let f_1 be a strategy for Player 1, and let $\rho = \mathsf{outcome}(f_1, \langle w, f_2\rangle) = s^0, s^1, s^2, \dots$. Let m be an index such that $s^m = s_0$ and for all $j > m$, we have that $s^j \neq s_0$. Since f_2 guarantees that s_0 is visited only finitely often, such an index m exists. By the definition of w, we have that $w(s^j, s^{j+1}) = -1$ for all $j > m$. Hence, Player 1 eventually (more precisely, in round $m + \mathcal{E}(\rho, m) + 1$ at the latest) runs out of energy. ☐

5 Memory Requirements in EGWU

Traditional EGs are memoryless: a player that has a winning strategy also has one that does not depend on the history of the play so far [5, 4]. As demonstrated in Example 1, this is not the case for EGWUs. Indeed, the described winning strategy f_1 of Player 1 may direct the token to different successors in different visits to v_0, according to the learned weight of $\langle v_0, v_0\rangle$, and it is not hard to see that a strategy that is independent of the learned weight does not win. In this section we study the memory requirements of winning strategies in EGWUs.

Consider an EGWU $G = \langle S_1, S_2, s_0, E, b, \tau\rangle$. A finite-memory structure for G is $\mathcal{M} = \langle M, m_0, \delta\rangle$, where M is a finite set of memory states, $m_0 \in M$ is an initial memory state, and $\delta : M \times (E \times \mathbb{Z}) \to M$ is a memory-update function. A memory structure is similar to an automaton with alphabet $E \times \mathbb{Z}$ that is executed in parallel to the game: it starts at m_0 and reads the edges traversed by the token, together with their learned weight: when the current memory state is m and the token traverses an edge e and the energy is updated by c, the new memory state is $\delta(m, \langle e, c\rangle)$.

A strategy for Player j that relies on $\mathcal{M}$ replaces the dependency on the history of the play by dependency on the current memory state of $\mathcal{M}$. In addition, the strategy depends on the current vertex of the game. Formally, a strategy for Player j is a function $f_j : M \times S_j \to S$. Then, when the current memory state is m and the token is in vertex $s \in S_j$, Player j moves the token to $s' = f_j(m, s)$. The strategy f_j is *memoryless* if it relies on a memory structure with a single memory state. A memoryless strategy for Player j is thus given by $f_j : S_j \to S$.

The outcome of strategies that rely on finite memory structures can be defined via the induced strategies that are defined with respect to histories. Here, we give a direct definition for the case the strategy of Player 2 is memoryless. Let $\pi = \langle f_1, \langle w, f_2 \rangle \rangle$ be a profile of strategies for Player 1 and Player 2, where f_1 relies on a memory structure $\mathcal{M} = \langle M, m_0, \delta \rangle$, thus $f_1 : M \times S_1 \to S$, and $f_2 : S_2 \to S$ is memoryless. Then, $\mathsf{outcome}(\pi)$ is the infinite play $\rho = s_0, w(s_0, s_1), s_1, w(s_1, s_2), s_2, \ldots \in S^\omega$ such that there is a sequences $\eta = m_0, m_1, m_2, \ldots \in M^\omega$ such that for all $i \geq 0$, if $s_i \in S_1$, then $s_{i+1} = f_1(m_i, s_i)$, and if $s_i \in S_2$, then $s_{i+1} = f_2(s_i)$. Also, $m_{i+1} = \delta(m_i, \langle (s_i, s_{i+1}), w(s_i, s_{i+1}) \rangle)$.

We can now analyse the memory required for the players in en EGWU. We start with Player 2, where things are easy.

Lemma 2. *Player 2 wins an EGWU G iff there is a weight function w that respects τ such that Player 2 wins G_w.*

Proof. Consider an EGWU $G = \langle S_1, S_2, s_0, E, b, \tau \rangle$. Assume first that there is a weight function w that respects τ such that Player 2 has a winning strategy f_2 in G_w. We claim that $\langle w, f_2 \rangle$ is a winning strategy for Player 2 in G. Intuitively, f_2 is winning when Player 1 knows w, all the more so when Player 1 does not know w. Formally, for every strategy f_1 for Player 1 in G, if $\mathsf{outcome}(f_1, \langle w, f_2 \rangle)$ is winning for Player 1 in G, then $\mathsf{outcome}(f_1', f_2)$ is winning for Player 1 in G_w, where f_1' is the strategy for Player 1 in G_w that is induced from f_1 by ignoring the learned weights. Hence, Player 2 wins G and we are done.

Assume now that for every weight function w that respects τ, Player 2 does not win G_w. Since traditional EGs are determined, the assumption implies that for every such weight function w, Player 1 has a winning strategy f_w in G_w. Consider an arbitrary strategy $\langle w, f_2 \rangle$ for Player 2 in G. We prove that $\langle w, f_2 \rangle$ is not a winning strategy for Player 2, implying that Player 2 does not win G.

By definition, the play $\mathsf{outcome}(f_w, \langle w, f_2 \rangle)$ in G coincides with the play $\mathsf{outcome}(f_w, f_2)$ in G_w. Since f_w is a winning strategy for Player 1 in G_w, the latter is winning for Player 1, and so $\langle w, f_2 \rangle$ is not a winning strategy for Player 2, and we are done.

Since for EGs, memoryless strategies are sufficient, Lemma 2 implies the following.

Theorem 4. *Player 2 wins an EGWU iff he has a memoryless winning strategy.*

Proof. Consider an EGWU $G = \langle S_1, S_2, s_0, E, b, \tau \rangle$. Intuitively, since Player 2 has full information, he can view the game as an EG, in which memoryless strategies are sufficient. Formally, assume Player 2 wins G. By Lemma 2, there is a weight function w that respects τ, such that Player 2 has a winning strategy f_2 in G_w. Since for EGs, memoryless strategies are sufficient, we may assume that f_2 is memoryless. As stated in the proof of Lemma 2, the strategy $\langle w, f_2 \rangle$ is winning for Player 2 in G.

We continue to Player 1, where things are more involved.

Theorem 5. *Player* 1 *wins an EGWU* G *iff he has a winning strategy with a memory of size exponential in* G*. The bound is tight and is required already in* 1*-controlled and flat EGWUs.*

Proof. We start with the upper bound. Consider an EGWU G and assume Player 1 wins G. By Lemma 1, he wins the EG $\mathsf{learn}(G)$, and can do so with a memoryless strategy. We show how a memoryless winning strategy for Player 1 in $\mathsf{learn}(G)$ induces a winning strategy for Player 1 in G that relies on a memory structure with state space $\mathcal{W}_\tau$. Since $|\mathcal{W}_\tau|$ is exponential in $|G|$, we are done.

Let $f_1' : V_1 \to V$ be a memoryless winning strategy for Player 1 in $\mathsf{learn}(G)$. Consider the memory structure $\mathcal{M} = \langle \mathcal{W}_\tau, g_0, \delta \rangle$, where for all $g \in \mathcal{W}_\tau$, $e \in E$, and $z \in \mathbb{Z}$, we define $\delta(g, \langle e, z \rangle) = \mathsf{update}(g, e, z)$. Now, consider the finite-memory strategy $f_1'' : \mathcal{W}_\tau \times S_1 \to S$ that relies on $\mathcal{M}$ and is defined, for every $g \in \mathcal{W}_\tau$ and $s \in S_1$, by $f_1''(g, s) = s'$, where the vertex $s' \in S$ is such that $f_1'(\langle s, g \rangle) = (\langle s, s' \rangle, g)$.

In Appendix A.1, we prove that f_1'' is winning in G.

We proceed to the lower bound and describe a family of EGWUs $G_1, G_2, \ldots$ such that for all $n \geq 1$, the game G_n is of size $O(n)$, all whose vertices are controlled by Player 1, it has a flat estimated weight function, Player 1 wins in G_n, yet needs a memory with at least 2^n memory states in order to win.

The EGWU $G_n = \langle S_1, S_2, v_0, E, n, \tau \rangle$ is described in Fig. 4. In addition to the weight estimation of the edges in the figure, we have, for every $0 \leq i \leq n-1$, that $\tau(\{\langle v_i, a_i \rangle, \langle v_i, b_i \rangle\}) = \langle -1, 0 \rangle$. Clearly, τ is flat. For a weight function $w : E \to \mathbb{Z}$ and $0 \leq i \leq n-1$, we say that w *blocks* a_i if $w(v_i, a_i) = -1$, and similarly for b_i. By the definition of τ, every weight function w that respects τ blocks at most one of a_i or b_i, and so there is an *open cycle* $C_w \subseteq E$, namely a cycle from v_0 back to itself all whose edges have weight 0.

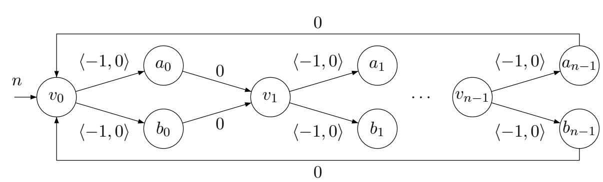

Fig. 4: The game graph G_n.

A winning strategy for Player 1 that is based on this observation first "scans" G_n in order to find an open cycle, and then traverses this cycle forever. Such a strategy has to remember for each $0 \leq i \leq n-1$, whether a_i is blocked, and thus requires space 2^n. In Appendix A.1, we describe the strategy formally, show that it is indeed winning, and that every winning strategy for Player 1 in G_n requires memory of size at least 2^n. □

6 Deciding Whether Player 1 Wins an EGWU

Let $G = \langle S_1, S_2, s_0, E, b, \tau \rangle$ be an EGWU. As proved in Lemma 1, Player 1 wins G iff he wins the EG $\mathsf{learn}(G)$. Since the size of $\mathsf{learn}(G)$ is exponential in $|G|$ and

EGs can be decided in polynomial time, this gives us an exponential upper bound for the problem of deciding whether Player 1 wins G. In this section, we show that we can do better and the problem of deciding whether Player 1 wins an EGWU is PSPACE-complete.

Our upper bound is based on the connection between G and $\mathsf{learn}(G)$. We first bound the diameter of $\mathsf{learn}(G)$ and then claim that the diameter is the computational bottleneck in the complexity of deciding EGs.

Lemma 3. *Consider an EGWU $G = \langle S_1, S_2, s_0, E, b, \tau\rangle$. The diameter of $\mathsf{learn}(G)$ is at most $2 \cdot |S| \cdot |E|$.*

Proof. Consider a simple path $p = v_0, v_1, \ldots v_n$ in $\mathsf{learn}(G)$, and let $p' = \langle s_0, g_0\rangle, \langle s_1, g_1\rangle, \ldots, \langle s_k, g_k\rangle \in (S \times \mathcal{W}_\tau)^*$ be the sequence obtained from p by removing vertices in $E_1 \times \mathcal{W}_\tau$. By the definition of the transitions of $\mathsf{learn}(G)$, for all $i \geq 1$ we have that $g_{i+1} = \mathsf{update}(g_i, e, c)$ for some $e \in E$ and $c \in Z$. Thus, $g_i \preceq g_{i+1}$. Hence, for every edge $e \in E$, once a weight for e is defined in a learned function that appears in p', it is fixed in all the successive learned functions. Formally, for every $i \geq 0$, if $g_i(e) \in \mathbb{Z}$, then $g_j(e) = g_i(e)$ for all $j \geq i$. Hence, at most $|E|$ different learned functions in $\mathcal{W}_\tau$ participate in p', and so $|p'| \leq |S| \cdot |E|$. Since vertices in $E_1 \times \mathcal{W}_\tau$ cannot be successive in p, it must be that $|p| \leq 2 \cdot |p'|$, and we are done. ☐

Theorem 6. *Let G be an EG of size exponential in n and diameter polynomial in n. Then, G can be decided in space polynomial in n.*

Proof. We describe an alternating Turing machine (ATM) that decides G in time polynomial in n. Since alternating polynomial time can be simulated in polynomial space [6], the bound follows. An ATM is a Turing machine whose states are partitioned into *existential* and *universal* states. A configuration of $\mathcal{T}$ describes its state, the content of the working tape, and the location of the reading head. A configuration is existential (universal) if the state of $\mathcal{T}$ in the configuration is existential (universal, respectively). A run of $\mathcal{T}$ is a tree in which each node corresponds to a configuration of $\mathcal{T}$: the root of the tree corresponds to the initial configuration; a node that corresponds to an existential configuration has a single successor, for one of the possible successor configurations; and a node that corresponds to a universal configuration has multiple successors, one for each possible successor configuration. The run is accepting iff all the branches of the tree reach an accepting configuration; that is, a configuration whose state is accepting.

Let $G = \langle S_1, S_2, s_0, E, b, w\rangle$. A *configuration* of G is a pair $\langle s, k\rangle \in (S \times \mathbb{N})^*$, describing a vertex in the game and the (non-negative) energy level of Player 1. The initial configuration is $\langle s_0, b\rangle$. Let d be the diameter of G. Recall that Player 1 wins G iff he has a strategy that guarantees that every outcome ρ satisfies $\mathcal{E}(\rho, n) \geq 0$, for all $n \geq 0$. Since Player 1 wins G iff he has a memoryless strategy that is winning in G, we can assume that for every outcome ρ of a winning strategy of Player 1, every cycle in ρ has a non-negative weight. Indeed, if a negative cycle is formed, Player 2 can force looping in it forever, which makes Player 1 lose.

We describe an ATM $\mathcal{T}$ that decides whether Player 1 wins G. The idea is that $\mathcal{T}$ simulates the game for d rounds. It does so by maintaining on its tape the number i of rounds taken so far, and the sequence $\langle s_0, k_0\rangle \cdot \langle s_1, k_1\rangle \cdot \ldots \langle s_i, k_i\rangle \in (S \times \mathbb{N})^*$ of configurations of the game so far.

As long as $i < d$, the ATM $\mathcal{T}$ proceeds as follows. Let $\langle s, k \rangle$ be the last configuration on the tape. If $s \in S_1$, the ATM $\mathcal{T}$ is in an existential configuration: it guesses a successor s' of s in G, and adds the configuration $\langle s', k' \rangle$ to the tape, with $k' = k + w(s, s')$. Then, $\mathcal{T}$ checks whether $k' < 0$, and if so, it rejects. Otherwise, it checks for every configuration $\langle s'', k'' \rangle$ on the tape whether $s' = s''$. If so, it accepts if $k' \geq k''$, and rejects otherwise. If s' has not been visited before, it increases i to $i + 1$ and moves to the next simulation step. If $s \in S_2$, the ATM proceeds similarly, except that now it is in a universal configuration, and thus the check is performed for every successor s' of s in G.

Since the diameter of G is d, all the computations of $\mathcal{T}$ terminate before i reaches d. Also, the length of the sequence of configurations that need to be checked is bounded by d, and since G is of size exponential in n, the description of each configuration is of polynomial length.

Since $\mathcal{T}$ accepts G iff Player 1 has a strategy that induces only non-negative cycles, while maintaining the energy above 0, it accepts G iff Player 1 wins. □

Theorem 7. *Deciding whether Player* 1 *wins a given EGWU is PSPACE-complete. PSPACE hardness holds already for flat EGWUs.*

Proof. We start with the upper bound. By Lemmas 1 and 3, deciding Player 1 wins a given EGWU G can be reduced to deciding an EG with of size exponential in $|G|$ and diameter polynomial in $|G|$. By Theorem 6, the latter can be decided in PSPACE.

For the lower bound, we describe a reduction from TQBF – the problem of determining the truth of quantified Boolean formulas. Let $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_n\}$ be sets of Boolean variables, and let φ be a Boolean propositional formula over the variables in $X \cup Y$. Also, let $\overline{X} = \{\overline{x_1}, \ldots, \overline{x_n}\}$, $\overline{Y} = \{\overline{y_1}, \ldots, \overline{y_n}\}$, and $L = X \cup \overline{X} \cup Y \cup \overline{Y}$. We assume that φ is in 3CNF. That is, $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$, where for all $1 \leq i \leq m$, it holds that $C_i = l_i^1 \vee l_i^2 \vee l_i^3$ for $l_i^1, l_i^2, l_i^3 \in L$. Finally, let $\theta = \exists x_1 \forall y_1 \exists x_2 \forall y_2 \ldots \exists x_n \forall y_n \varphi$. By [20], deciding whether θ is valid is PSPACE-hard.

Given θ, we define an EGWU G_θ such that θ is valid iff Player 1 wins G_θ. The idea of the reduction is as follows. A play in $\mathcal{G}_\theta$ starts in an *assignment part*, where the players choose assignment to the variables: Player 1 to the variables in X, and Player 2 to the variables in Y. Then, the play moves to a *check part*, where Player 2 challenges the assignment induced form the outcome in the assignment part: Player 2 chooses a clause C_i, for $1 \leq i \leq m$, and Player 1 wins if it can respond with a literal in $\left\{l_i^1, l_i^2, l_i^3\right\}$ that was assigned T in the assignment part. The challenging part in the reduction is to define the estimated weight function such that Player 1 can indeed respond only with literals that were assigned T.

We now explain the reduction in detail. In Fig. 5 we describe the assignment part of G_θ. In the figures for both parts, black (undashed) edges are known to have weight 0 and dashed edges e have $\tau(e) = \langle -1, 0 \rangle$, and thus can be assigned either -1 or 0. Additional constraints by τ are defined below.

Note that a strategy for Player 1 and a token strategy for Player 2 induce an assignment to the variables in $X \cup Y$: traversing the edge $\langle a_i, a_i^+ \rangle$ (respectively, $\langle a_i, a_i^- \rangle$) corresponds to assigning T (respectively, F) to the variable x_i, and similarly for the edges leaving the vertex b_i and the variable y_i.

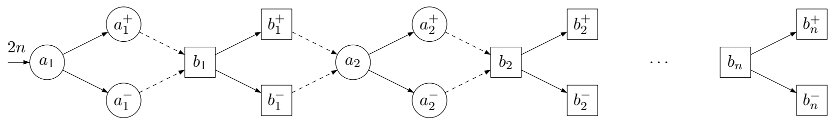

Fig. 5: The assignment part of G_θ.

Note that since the initial energy is $2 \cdot n$ and no matter how the players proceed the token traverses $2 \cdot n$ dashed edges, namely ones that can get weight -1, then for every strategy of Player 1, there is a strategy of Player 2 such that the energy level of Player 1 after the play completes the assignment part is 0.

In Fig. 6 we describe the check part of G_θ. The figure corresponds to the case $C_1 = x_1 \vee \overline{y_1} \vee \overline{x_2}$, $C_2 = \overline{x_1} \vee y_1 \vee \overline{x_2}$, and $C_m = \overline{y_1} \vee x_2 \vee \overline{y_n}$. Black edges are known with weight 0, while other edges are labeled with $\langle -1, 0 \rangle$. Note that the vertices b_n^+ and b_n^- appear in both figures, thus a play enters the check part immediately after the players fix an assignment to the variables.

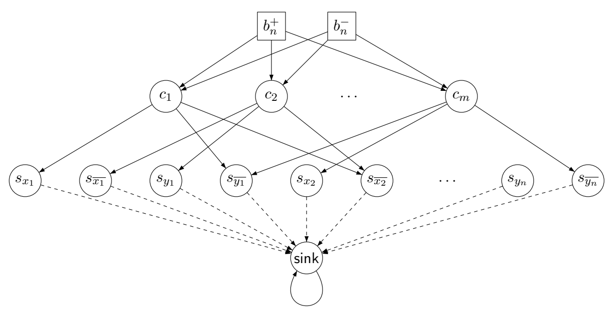

Fig. 6: The check part of G_θ.

In the check part, Player 2 challenges this assignment by choosing a clause C_i, for $1 \leq i \leq m$. He does so by moving the token to the *clause vertex* c_i. Then, Player 1 responds by moving the token to a *literal vertex* s_l, for a literal l that appears in C_i. From s_l, the token continues to the sink, where it can loop for free. The edge from s_l to the sink may have weight -1, and so, keeping in mind that Player 1 may have already spent his initial energy in the check part, his goal is to choose a literal l for which a weight function that respects τ cannot assign -1 to the edge $\langle s_l, \mathsf{sink} \rangle$.

The definition of τ guarantees that Player 1 can do so iff θ is valid: For each literal $l \in L$, let E_l be the following set of two edges: the edge that is traversed when l is assigned $\top$ in the assignment part, and the edge from s_l to the sink in the check part. For example, $E_{x_1} = \{\langle a_1, a_1^+ \rangle, \langle s_{x_1}, \mathsf{sink} \rangle\}$ and $E_{\overline{y_2}} = \{\langle b_2, b_2^- \rangle, \langle s_{\overline{y_2}}, \mathsf{sink} \rangle\}$. By adding the requirement $\tau(E_l) = \langle -1, 0 \rangle$, it is guaranteed that the weight of at least one of the edges is 0. Intuitively, these requirements imply that Player 1 can traverse the edge from s_l to the sink for free iff Player 1 has already spent energy on assigning $\top$ to l in the check part, which is possible iff θ is valid. Note that G_θ is flat.

In Appendix A.2, we prove that θ is valid iff Player 1 wins G_θ. $\square$

In the *unknown initial-energy* problem we ask, given an EGWU $G = \langle S_1, S_2, s_0, E, b', \tau\rangle$, whether there is an initial energy b such that Player 1 wins in $G_b = \langle S_1, S_2, s_0, E, b, \tau\rangle$. Thus, rather than checking whether Player 1 wins an EGWU with a given initial energy, we ask for the existence of a finite initial energy with which Player 1 can win, and we seek to find a minimal such initial energy.

By Lemma 1, Player 1 wins an EGWU G iff he wins the EG $\mathsf{learn}(G)$. Moreover, the proof shows that winning G and $\mathsf{learn}(G)$ can be done with the same initial energy. Thus, solving the unknown initial-energy problem for G can be reduced to solving it for $\mathsf{learn}(G)$. By [5], the minimal energy required for winning an EG G' is at most $-\sum_{e\in E'} \min\{0, w(e)\}$, where E' and w are the edges and weight function of G'. This gives us a bound polynomial in the size of $\mathsf{learn}(G)$, which is exponential in $|G|$. As we shall see below, a tighter analysis leads to a bound that is only polynomial in $|G|$. Essentially (see full proof in Appendix A.3), the bound follows from the fact winning strategies of Player 1 avoid negative cycles. Thus, the important parameter is not the size of $\mathsf{learn}(G)$ but its diameter.

Lemma 4. *Consider an EGWU $G = \langle S_1, S_2, s_0, E, b', \tau\rangle$. There is $b \in \mathbb{N}$ such that Player 1 wins G_b iff Player 1 wins $G_{\hat{b}}$, for $\hat{b} = 2 \cdot |S| \cdot |E| \cdot l_G$. The bound on $\hat{b}$ is tight.*

By Lemma 4, we can solve the unknown initial-energy problem for an EGWU G by checking whether Player 1 wins $G_{\hat{b}}$, for $\hat{b} = 2 \cdot |S| \cdot |E| \cdot l_G$. Moreover, a minimal sufficient initial energy can be searched, and so both the decision and optimization problems can be solved in PSPACE.

7 Deciding Whether Player 2 Wins an EGWU

Theorem 8. *Deciding whether Player 2 wins a given EGWU is NP-complete. Hardness in NP holds already for 1-controlled and global EGWUs.*

Proof. For the upper bound, consider a non-deterministic Turing machine that given an EGWU $G = \langle S_1, S_2, s_0, E, b, \tau\rangle$, guesses a weight function w that respects τ, and accepts iff Player 2 has a winning strategy in G_w. By Lemma 2, this machine decides whether Player 2 wins G. Since traditional energy games can be decided in polynomial time when weights are given in unary [5], we are done.

For the lower bound, we describe a reduction from the Vertex-Cover problem. Given an undirected graph $G = \langle V, E\rangle$ and $k \in \mathbb{N}$, we construct an EGWU G' such that G has a vertex cover of size at most k iff Player 2 wins G'. The reduction is illustrated in Fig. 7. In the figure, dashed edges e have $\tau(e) = \langle -1, 0\rangle$ and black edges are known with weight 0. An additional vertex s_0 has black edges to all vertices of the form v_{in}, which we omit from clarity.

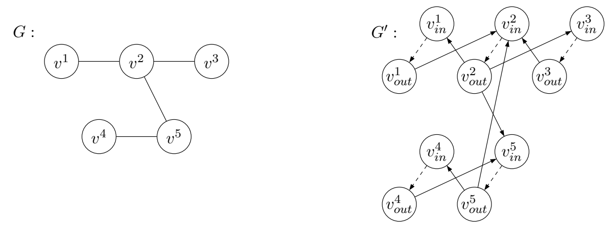

Fig. 7: The graph G has a vertex cover of size k iff Player 2 wins G'.

We define $G' = \langle S_1, \varnothing, s_0, E, 0, \tau\rangle$ as follows. First, $S_1 = \{v_{in}, v_{out} : v \in V\} \cup \{s_0\}$. Thus, each vertex in G has two copies in S_1: an entry copy and an exit copy. The exit copy is going to be reachable from the entry copy via an *internal edge* that may have a negative weight -1. Also, each edge $\langle v, u\rangle$ in G induces a cycle that involves the two internal edges of v and u, and is reachable from the initial vertex s_0. The estimated weight function τ restricts Player 2 to allocate a negative weight only to k internal edges. Since all cycles are reachable from s_0, Player 2 can prevent all the strategies of Player 1 from finding a non-negative cycle iff for all edges $\langle v, u\rangle$ in G, at least one of the internal edges of v and u have weight -1, which is possible iff G has a vertex cover of size k.

Formally, $E' = E_{init} \cup E_{internal} \cup E_{orig}$, where $E_{init} = \{\langle s_0, v_{in}\rangle : v \in V\}$ connects s_0 to all the entry copies, $E_{internal} = \{\langle v_{in}, v_{out}\rangle : v \in V\}$ connects each entry copy to the corresponding exit copy, and $E_{orig} = \{\langle v_{out}, u_{in}\rangle : \{v, u\} \in E\}$ causes each edge $\langle v, u\rangle \in E$ to induce a cycle $(v_{in}, v_{out}, u_{in}, u_{out})$ in G'.

The function τ is such that for all $e \in E_{internal}$, we have that $\tau(e) = \langle -1, 0\rangle$. All other edges e have $\tau(e) = 0$. Finally, $\tau(E) = \langle -k, 0\rangle$. Note that G' is global.

In Appendix A.4, we prove that G has a vertex cover of size at most k iff Player 2 wins G'. Essentially, a vertex cover of size at most k induces a weight function w that assigns -1 to at least one internal edge in every pair of successive internal edges, causing all plays to run out of energy. On the other hand, if no vertex cover of size at most k exists, then every strategy for Player 2, namely a weight function w that respects τ, leaves in G' at least one cycle of weight 0. Then, a play that reaches the cycle and loops in it forever is winning for Player 1, and so Player 1 has a strategy that wins against w. ☐

Recall the unknown initial-energy problem, where we ask whether Player 1 can win an EGWU with some initial value. In the dual problem we ask, given $G = \langle S_1, S_2, s_0, E, b', \tau\rangle$, whether there is a strategy for Player 2 that is winning in G_b for all $b \in \mathbb{N}$. It is not hard to see that such a strategy exists iff there is a weight function w that respects τ and a token strategy f_2 that forces a cycle with a negative weight in the EG G_w, where we can restrict attention to memoryless token strategies. Given w and a memoryless strategy f_2 in G_w, consider the weighted graph $G_w^{f_2} = \langle S, E_{f_2}, w\rangle$ obtained by applying $\langle w, f_2\rangle$ to G. That is, edges are labeled according to w, and each vertex $s \in S_2$ has a single successor, which is $f_2(s)$. The paths in $G_w^{f_2}$ correspond to all the possible outcomes of a play in G when Player 2 uses the strategy $\langle w, f_2\rangle$. Thus, the latter is a solution to the dual unknown initial-energy problem iff all cycles in $G_w^{f_2}$ are negative. Hence, the problem can be solved in NP by guessing w and f_2 and then checking, in polynomial time, whether $G_w^{f_2}$ has a non-negative cycle. Since the EGWU G' described in the proof of Theorem 8 is such that Player 1 does not win G' iff he does not win G'_b for all $b \in \mathbb{N}$, the reduction from the vertex-cover problem applies also here, and so the dual problem is NP-complete.

8 Discussion

We introduced and studied a new type of uncertainty in qualitative games: energy games with uncertainty about weights. As elaborated in Section 1, EGWU can model a variety of applications in which the exact cost of actions is only estimated. Unlike uncertainty about location, information about weights is fixed in an offline manner and is revealed during the play. This makes weight uncertainty very different from location uncertainty, and we study their theoretical properties and the complexity of decision problems for EGWU.

Possible directions for future research include conceptual as well as technical extensions. Conceptual extensions involve variants of the model, and require further research. Such variants include, for example, uncertainty about both location and weights, bounds on the memory of the system (and hence, bound on the learning that can be performed), a stochastic setting and randomized strategies (which also enable a more informative competitive analysis of the price of uncertainty), and a setting with energy constraints, possibly with uncertainty, also for the environment.

On the more technical side, an interesting extension is to EGWU with *parity* winning conditions. Each parity condition α defines a subset of S^ω, namely the set of plays that satisfy α. In order to win an EG augmented with a parity condition α, Player 1 has to generate an outcome that satisfies α and does not run out of energy. Clearly, all our negative results for EGWUs apply also to parity EGWUs. Also, it is not hard to see that our construction of $\mathsf{learn}(G)$ from G applies also to parity EGWUs, with $\alpha' = \alpha \times \mathcal{W}_\tau$. Thus, Player 1 wins a parity-EGWU G iff he wins the parity-EG $\mathsf{learn}(G)$. Since the learning performed in the proof of Theorem 2 is along a finite prefix, it applies also to parity EGWUs. As for decision problems, deciding whether Player 1 wins a parity-EGWU G can be reduced to deciding the parity EG $\mathsf{learn}(G)$, and deciding whether Player 2 wins can be reduced to solving the parity-EG G_w for a guessed weight function w. Beyond the complexity of deciding parity-EGs ([7] solved the unknown initial-energy variant), it may well be that a naive application of the above reductions is not optimal. Indeed, for EGWUs we were able to improve it, for example by an analysis that based on the diameter of $\mathsf{learn}(G)$ rather than it size. Thus, the exact complexity of solving parity-EGWUs requires further research.

References

1. G. Amram, S. Maoz, O. Pistiner, and J. O. Ringert. Efficient algorithms for omega-regular energy games. In *24th International Symposium on Formal Methods*, volume 13047 of *Lecture Notes in Computer Science*, pages 163–181. Springer, 2021.
2. D. Berwanger, K. Chatterjee, M. De Wulf, L. Doyen, and T.A. Henzinger. Strategy construction for parity games with imperfect information. *Information and computation*, 208(10):1206–1220, 2010.
3. R. Bloem, K. Chatterjee, and B. Jobstmann. Graph games and reactive synthesis. In *Handbook of Model Checking.*, pages 921–962. Springer, 2018.

4. P. Bouyer, U. Fahrenberg, K.G. Larsen, N. Markey, and J. Srba. Infinite runs in weighted timed automata with energy constraints. In *6th International Conference on Formal Modeling and Analysis of Timed Systems*, volume 5215 of *Lecture Notes in Computer Science*, pages 33–47. Springer, 2008.
5. A. Chakrabarti, L. de Alfaro, T.A. Henzinger, and M. Stoelinga. Resource interfaces. In *International Workshop on Embedded Software*, pages 117–133. Springer, 2003.
6. A.K. Chandra, D.C. Kozen, and L.J. Stockmeyer. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133, 1981.
7. K. Chatterjee and L. Doyen. Energy parity games. *Theoretical Computer Science*, 458:49–60, 2012.
8. K. Chatterjee, L. Doyen, T. A. Henzinger, and J-F. Raskin. Algorithms for ω-regular games with imperfect information. In *Proc. 15th Annual Conf. of the European Association for Computer Science Logic*, volume 4207 of *Lecture Notes in Computer Science*, pages 287–302, 2006.
9. K. Chatterjee, L. Doyen, T.A. Henzinger, and J-F. Raskin. Generalized mean-payoff and energy games. In *Proc. 30th Conf. on Foundations of Software Technology and Theoretical Computer Science*, volume 8 of *LIPIcs*, pages 505–516. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010.
10. K. Chatterjee, A.K. Goharshady, and Y. Velner. Quantitative analysis of smart contracts. In *27th European Symposium on Programming Languages and Systems*, volume 10801 of *Lecture Notes in Computer Science*, pages 739–767. Springer, 2018.
11. A. Degorre, L. Doyen, R. Gentilini, J. Raskin, and S. Torunczyk. Energy and mean-payoff games with imperfect information. In *Proc. 19th Annual Conf. of the European Association for Computer Science Logic*, pages 260–274, 2010.
12. L. Doyen and J-F. Raskin. Games with imperfect information: theory and algorithms. *Lectures in Game Theory for Computer Scientists*, 10, 2011.
13. T.A. Henzinger. From Boolean to quantitative notions of correctness. In *Proc. 37th ACM Symp. on Principles of Programming Languages*, pages 157–158, 2010.
14. O. Kupferman and N. Shenwald. Games with trading of control. In *Proc. 34th Int. Conf. on Concurrency Theory*, volume 279 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.
15. O. Kupferman and M.Y. Vardi. Synthesis with incomplete informatio. In *Advances in Temporal Logic*, pages 109–127. Kluwer Academic Publishers, 2000.
16. D.A. Martin. Borel determinacy. *Annals of Mathematics*, 65:363–371, 1975.
17. G. A. Pérez. The fixed initial credit problem for partial-observation energy games is ack-complete. *Information processing letters*, 118:91–99, 2017.
18. A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proc. 16th ACM Symp. on Principles of Programming Languages*, pages 179–190, 1989.
19. J.H. Reif. The complexity of two-player games of incomplete information. *Journal of Computer and Systems Science*, 29:274–301, 1984.
20. L.J. Stockmeyer and A.R. Meyer. Word problems requiring exponential time. In *Proc. 5th ACM Symp. on Theory of Computing*, pages 1–9, 1973.
21. M.Y. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *Journal of Computer and Systems Science*, 32(2):182–221, 1986.

A Missing Proofs

A.1 Missing details in the proof of Theorem 5

We start with the upper bound and prove that f_1'' is winning in G. Consider a memory structure $\mathcal{M} = \langle M, m_0, \delta \rangle$. We extend δ to $\delta^* : M \times (E \times \mathbb{Z})^* \to M$ in the expected way: For all $m \in M$, we have that $\delta^*(m, \epsilon) = m$, and for all $m \in M$, $p \in (E \times \mathbb{Z})^*$, $e \in E$, and $c \in \mathbb{Z}$, we have that $\delta^*(m, p \cdot \langle e, c\rangle) = \delta(\delta^*(m, p), \langle e, c\rangle)$.

Now, back to f_1''. In Lemma 1, we showed how f_1' induces a winning strategy $f_1 : (S \times \mathbb{Z})^* \cdot S_1 \to S$ for Player 1 in G. In the strategy f_1, for every history $h \cdot s \in (S \times \mathbb{Z})^* \cdot S_1$, we have that $f(h \cdot s) = s'$, where $s' \in S$ is such that $f_1'(\langle s, g_{h\cdot s}\rangle) = \langle (s, s'), g_{h\cdot s}\rangle$. Consider a history $h \cdot s \in (S \times \mathbb{Z})^* \cdot S_1$ in G. By the definition of $\mathcal{M}$, we have that $\delta^*(g_0, h) = g_h$. Hence, $f_1''(g_{h\cdot s}, s) = f_1(h \cdot s)$, and so f_1'' coincides with f_1. Since f_1 is winning for Player 1 in G, so is f_1'', and we are done.

We continue to the lower bound, describe formally a strategy for Player 1 that is winning in G_n, and prove that every winning strategy for Player 1 in G_n requires memory of size at least 2^n.

Consider the following strategy f_1 for Player 1: for all $0 \le i \le n-1$, proceed from v_i to v_{i+1} through a_i, and learn whether a_i is blocked. After returning to v_0, perform the following cycle C forever: for each $0 \le i \le n-1$, if a_i is not blocked, then proceed to v_{i+1} through a_i. Otherwise, in which case b_i is must not be blocked, proceed to v_{i+1} through b_i. Since all the edges in C have weight 0, and the first cycle that Player 1 performs costs at most n, which is the initial energy of Player 1, the strategy f_1 is winning.

We now show that every winning strategy for Player 1 in G_n requires memory of size at least 2^n. Intuitively, Player 1 has to remember, for each $0 \le i \le n-1$, whether a_i is blocked.

For every $A \subseteq \{0, \ldots, n-1\}$, consider the weight function w_A that for each $0 \le i \le n-1$, blocks a_i iff $i \notin A$ and blocks b_i iff $i \in A$. For all edges $e \in E$ that are not blocked, we define $w_A(e) = 0$. Note that w_A respects τ. For $A \subseteq \{0, \ldots, n-1\}$, let C_A be the only open cycle in G_n with weights in w_A. That is, for all $0 \le i \le n-1$, the cycle C_A goes through a_i if $i \in A$, and goes through b_i otherwise. Note that every cycle $C \ne C_A$ has $\mathcal{E}_\Delta(C) < 0$.

Let f_1 be a finite-memory strategy for Player 1 that relies on a memory structure $\mathcal{M} = \langle M, m_0, \delta\rangle$, and assume $|M| < 2^n$. Note that since $S_2 = \varnothing$, a strategy for Player 2 is a weight function that respects τ. We argue that f_1 is not a winning strategy for Player 1. Specifically, there is a strategy for Player 2, namely a weight function w_A, such that $\mathsf{outcome}(f_1, w_A)$ is not winning for Player 1.

Assume by way of contradiction that for every weight function w_A, it holds that $\mathsf{outcome}(f_1, w_A)$ is winning for Player 1. Since the energy level of Player 1 cannot be increased during the game, he can traverse cycles different from C_A only finitely often, and so $\mathsf{outcome}(f_1, w_A)$ must have a suffix $r_A = (\langle v_0, 0\rangle, \langle x_0, 0\rangle, \langle v_1, 0\rangle, \langle x_1, 0\rangle \ldots, \langle v_{n-1}, 0\rangle, \langle x_{n-1}, 0\rangle)^\omega$, where $x_i \in \{a_i, b_i\}$ and $v_0, x_0, v_1, x_1, \ldots, v_{n-1}, x_{n-1} = C_A$. Let p_A be a prefix such that $\mathsf{outcome}(f_1, w_A) = p_A \cdot (r_A)^\omega$. Let $\delta^*(m_0, p_A) = m_A$. That is, after traversing p_A, the memory state of Player 1 is m_A. Since $|M| < 2^n$, there must be two different sets $A_1, A_2 \subseteq \{0, \ldots, n-1\}$ such that $A_1 \ne A_2$ with $m_{A_1} = m_{A_2} = m$.

Let $C_{A_1} = v_0, x_0^1, v_1, x_1^1, \ldots, v_{n-1}, x_{n-1}^1$ and $C_{A_2} = v_0, x_0^2, v_1, x_1^2, \ldots, v_{n-1}, x_{n-1}^2$, where for all $0 \le i \le n-1$, we have that $x_i^1, x_i^2 \in \{a_i, b_i\}$. On the one hand, as $A_1 \ne A_2$, there must an index $0 \le j \le n-1$ such that $j \in A_1 \setminus A_2$ or $j \in A_2 \setminus A_1$, implying that $x_j^1 \ne x_j^2$. On the other hand, we prove that for all $0 \le i \le n-1$, it must be that $x_i^1 = x_i^2$, contradicting the existence of A_1 and A_2 as above.

Intuitively, since f_1 relies om $\mathcal{M}$, the continuation of the play is determined by the current vertex of the game and the current memory state of $\mathcal{M}$, which are v_0 and m, and also by the weights to be revealed, which are all zeros in both C_{A_1} and C_{A_2}.

The proof proceeds by an induction on i. For the induction base, recall that $\delta^*(m_0, p_{A_1}) = \delta^*(m_0, p_{A_2}) = m$. Hence, $x_0^1 = x_0^2 = f_1(m, v_0)$. For the induction step, assume that $v_0, x_0^1, v_1, x_1^1, \ldots, v_{j-1}, x_{j-1}^1, v_j = v_0, x_0^2, v_1, x_1^2, \ldots, v_{j-1}, x_{j-1}^2, v_j$ for $0 \leq j \leq n-2$. Denote $\langle v_0, 0\rangle, \langle x_0^1, 0\rangle, \langle v_1, 0\rangle, \langle x_1^1, 0\rangle, \ldots, \langle v_{j-1}, 0\rangle, \langle x_{j-1}^1, 0\rangle, v_j$ by h_j. Let $m^j = \delta^*(m, h_j)$ be the memory state that M reaches after traversing h_j from m. Then, $x_{j+1}^1 = x_{j+1}^2 = f_1(m^j, v_j)$, and so the claim holds also for $j+1$, and we are done.

A.2 Correctness of the reduction in Theorem 7

We now prove that θ is valid iff Player 1 wins G_θ. Assume first that θ is valid. For $0 \leq i \leq n$, let $\mathcal{A}_i = \{\mathsf{F}, \mathsf{T}\}^{\{x_1, \ldots, x_i, y_1, \ldots, y_i\}}$ be the set of possible assignments to the first i variables in X and Y. In particular, $\mathcal{A}_0 = \varnothing$. For an assignment $\alpha_X : \{\mathsf{F}, \mathsf{T}\}^X$ and $0 \leq i \leq n-1$, let $\alpha_X^i : \{\mathsf{F}, \mathsf{T}\}^{\{x_1, \ldots, x_i\}}$ be the restriction of α_X to the first i variables in X. Assignments α_Y^i are defined similarly, with respect to an assignment $\alpha_Y : \{\mathsf{F}, \mathsf{T}\}^Y$.

Since θ is valid, there is a strategy $\gamma : \mathcal{A}_0 \cup \ldots \cup \mathcal{A}_{n-1} \to \{\mathsf{F}, \mathsf{T}\}$, such that for every assignment $\alpha_Y \in \{\mathsf{F}, \mathsf{T}\}^Y$, the assignment $\alpha_X \in \{\mathsf{F}, \mathsf{T}\}^X$ that assigns values to the variables in X according to the suggestion of γ, is such that $\alpha_X \cup \alpha_Y$ satisfies φ. Formally, for all $0 \leq i \leq n-1$, we have that $\alpha_X(x_{i+1}) = \gamma(\alpha_Y^i \cup \alpha_X^i)$. Note that $\alpha_X(x_{i+1})$ only depends on the assignment to the first i variables in X and Y.

Consider a play in G. Let $\alpha_Y \in \{\mathsf{F}, \mathsf{T}\}^Y$ be the assignment induced from the transitions Player 2 takes from the vertices $b_1, \ldots, b_n$. That is, if Player 2 traverses the edge $\langle b_i, b_i^+\rangle$, then $\alpha_Y(y_i) = \mathsf{T}$, and if Player 2 traverses the edge $\langle b_i, b_i^-\rangle$, then $\alpha_Y(y_i) = \mathsf{F}$. Then, let $\alpha_X \in \{\mathsf{F}, \mathsf{T}\}^X$ be the assignment induced by γ and α_Y. Now, consider the strategy f_1 for Player 1 in G that behaves as follows:

- **(1)** At a vertex a_i, move the token to a vertex that corresponds to $\alpha_X(x_i)$. That is, if $\alpha_X(x_i) = \mathsf{T}$, then move the token to a_i^+, and if $\alpha_X(x_i) = \mathsf{F}$, then move the token to a_i^-.
- **(2)** At a clause vertex c_j, if there is a literal $z \in \{l_j^1, l_j^2, l_j^3\}$ for which an edge in E_z has been traversed in the assignment part, then move the token to the sink via s_z. If no such literal exists, move the token to the sink via an arbitrary literal vertex.

We prove that f_1 is winning. Consider a weight function w that respects τ, and a token component f_2. Let $\rho = \mathsf{outcome}(f_1, \langle w, f_2\rangle)$. Note that ρ includes exactly $2 \cdot n$ transitions in the assignment part. We distinguish between two cases. First, if ρ traverses in the assignment part an edge e with $w(e) = 0$, then the energy level of Player 1 when the token reaches a clause vertex c_j is at least 1, and so he can reach the sink with a non-negative energy level no matter which literal vertex s_z is chosen in Step (2). Second, if all the edges e that ρ traverses in the assignment part have $w(e) = -1$, in which case the energy level of Player 1 when the token reaches a clause vertex c_j is 0, we prove that the literal z specified in Step (2) exists. Then, the restriction on $\tau(E_z)$ guarantees that the weight on the edge from s_z to the sink is 0, and so Player 1 can reach the sink with energy level 0 and stay there forever. In order to see that the required literal z exists, recall that the assignment $\alpha_X \cup \alpha_Y$ satisfies φ. Hence, for every clause C_j, there is a literal $z \in \{l_j^1, l_j^2, l_j^3\}$ such that $(\alpha_X \cup \alpha_Y)(z) = \mathsf{T}$, and so ρ traversed an edge in E_z in the assignment part.

For the second direction, assume that Player 1 has a winning strategy f_1 in G_θ. We show that θ is valid. Let $\alpha_Y \in \{\mathsf{T},\mathsf{F}\}^Y$ be an assignment to the variables in Y. We construct an assignment $\alpha_X \in \{\mathsf{T},\mathsf{F}\}^X$ to the variables in X as follows. For every $1 \leq i \leq n$, let h_i be the history of the play ending in a_i when Player 1 follows f_1, Player 2 proceeds according to α_Y, and all weights revealed (except for edges with known weight 0) are -1.

Formally, let x_j be b_j^+ if $\alpha_Y(y_j) = \mathsf{T}$, and b_j^- otherwise, for all $j < i$. Then, $h_1 = a_1$, and for all $i > 2$ we have that $h_i = \langle a_1, -1\rangle, \langle f(h_1), 0\rangle, \langle b_1, -1\rangle, \langle x_1, 0\rangle, \langle a_2, -1\rangle, \langle f(h_2), 0\rangle, \langle b_2, -1\rangle, \langle x_2, 0\rangle, \ldots \langle a_{i-1}, -1\rangle, \langle f(h_{i-1}), 0\rangle, \langle b_{i-1}, -1\rangle, \langle x_{i-1}, 0\rangle, a_i$. Notice, that $h_n \cdot \langle a_n, -1\rangle, \langle f(a_n), 0\rangle, \langle b_n, -1\rangle, x_n$ is a prefix of a play in G_θ that is induced by f_1, and reaching the start of the check part (i.e., the vertex $x_n \in \{b_n^+, b_n^-\}$) with energy level 0. Since f_1 is winning, for every $1 \leq j \leq m$ there is a literal l that appears in C_j such that the edge corresponding to l in E_l has been traversed in the assignment part. Indeed, otherwise *Player* 2 can choose to proceed from x_n to a vertex c_k where all literal l_k^1, l_k^2, l_k^3 are such that their corresponding edges in the assignment part has not been traversed. Weighting those edges with 0, and the edges going out from c_k with -1, is weighting that respects τ and making Player 1 lose while using f_1, in contradiction to f_1 being winning.

Thus, for every $1 \leq j \leq m$, there is a literal l that appears in C_j such that the edge corresponding to l in E_l has been traversed in the assignment part. Thus, the assignment α_X that corresponds to f_1 (that is, $\alpha_X(x_i)$ is T if $f_1(h_i) = a_{i+1}^+$ and is F otherwise) is such that $\alpha_X \cup \alpha_Y$ satisfies φ.

A.3 Proof of Lemma 4

Proof. Assume that there is $b \in \mathbb{N}$ such that Player 1 wins the EGWU G_b. Let f_1 be a memoryless winning strategy for Player 1 the EG $\mathsf{learn}(G_b)$. By Lemma 1 and [4, 5], such a strategy exists. Thus, every path p induced by f_1 is such that every cycle C in p satisfies $\mathcal{E}_\Delta(C) \geq 0$. Let $m \in \mathbb{N}$ be such that $\mathcal{E}(p, m)$ is minimal. Since f_1 is winning, $\mathcal{E}(p, n) \geq 0$ for every $n \in \mathbb{N}$, and thus the minimal value is well defined. Note that Player 1 can traverse p with initial energy $\max\{0, -\mathcal{E}(p, m)\}$. Let p' be the prefix of p up to the m-th position. Thus, $\mathcal{E}_\Delta(p') = \mathcal{E}(p, m)$. Now, let p'' be the simple path obtained from p' by removing all cycles. Since all the cycles in p are non-negatives, we have that $\mathcal{E}_\Delta(p'') \leq \mathcal{E}_\Delta(p') = \mathcal{E}(p, m)$. Also, since p'' is simple, then, by Lemma 3, we have that $\mathcal{E}_\Delta(p'') \geq -2 \cdot |S| \cdot |E| \cdot l_G$. Thus, $\mathcal{E}(p, m) \geq -2 \cdot |S| \cdot |E| \cdot l_G$, and so $2 \cdot |S| \cdot |E| \cdot l_G$ is a sufficient initial energy for Player 1 to win in $\mathsf{learn}(G)$, and thus also in G.

As for tightness, the family of EGWUs $G_{n,l}$ described in the proof of Theorem 2 is such that in $G_{n,l}$ both $|S|$ and $|E|$ are of size $O(n)$, $l_G = l$, and the minimal required energy is $O(l \cdot n^2)$. □

A.4 Correctness of the reduction in Theorem 8

Assume first that G has a vertex cover $C \subseteq V$ of size at most k. Consider the weight function $w_C : E' \to \mathbb{Z}$ where for all $v \in V$, we have that $w_C(v_{in}, v_{out}) = -1$ if

$v \in C$, and $w_C(v_{in}, v_{out}) = 0$ otherwise. All the other edges in E' have weight 0. Since $|C| \leq k$, then $w(E_{internal}) \in [-k, 0]$, and so w_C respects τ. Consider now a strategy f_1 for Player 1, and let ρ be the outcome of f_1 and w_C. Note that every second edge in ρ is internal. Since C is a vertex cover, then for all successive internal edges, at least one of the has weight -1. Since all other edges have weight 0, then $\mathcal{E}_\Delta(p) < 0$ for a prefix of length at most 4 of ρ. Thus, w_C is a winning strategy for Player 2.

Assume now that G has no vertex cover of size at most k. Consider the weight function $w : E' \to \mathbb{Z}$, let $C_w \subseteq E_{internal}$ be such that for all $e \in E_{internal}$, we have that $e \in C_w$ iff $w(e) = -1$. Since $\tau(E_{internal}) = \langle -k, 0 \rangle$, we have that $|C_w| \leq k$. Hence, as G has no vertex cover of size at most k, there is an edge $\langle v, u \rangle \in E$ such that neither $w_C(v_{in}, v_{out}) = -1$ nor $w_C(u_{in}, u_{out}) = -1$. Therefore, $w_C(v_{in}, v_{out}) = w_C(u_{in}, u_{out}) = 0$, and so the strategy f_1 of Player 1 that moves the token to v_{in} and then cycles forever in $v_{in}, v_{out}, u_{in}, u_{out}$ is such that $\mathsf{outcome}(f_1, w)$ is winning for Player 1. Thus, w is not a winning strategy for Player 2, and we are done.