Introduction

The **Game of Atlas** is a word-based game that can be represented as a **directed graph**, where:

- **Nodes represent countries**.
- **Edges exist if the last letter of one country matches the first letter of another**.

Using this representation, we can apply **graph theory and deep learning techniques** to **analyze, optimize, and strategize our gameplay**. This report outlines **four strategic approaches**:

1. **Out-Degree Differential Strategy** – Restricting the opponent's viable moves.
2. **Entropy Minimization and Forced Predictability** – Forcing opponents into predictable patterns.
3. **Centralized Choke-Point Strategy** – Controlling key transition nodes.
4. **Local Lookahead Strategy** – Simulating multi-step moves for long-term dominance.

Additionally, we explore **two community detection techniques**:

1. **Infomap**, an information-theoretic approach that models the game as an **information flow system**.
2. **GNN + HDBSCAN**, a deep learning-based method that learns **graph embeddings and clusters countries dynamically**.

Each strategy was **simulated in a 20-move game**, with results visualized through **performance metrics and interactive graphs**.

Task 1: Competitive Advantage Strategies

1. Out-Degree Differential Strategy

> **Goal:** Maximize your own move options while ensuring that the opponent has significantly fewer viable responses.

Failed Sub-Strategy: Maximizing Your Own Out-Degree

Idea:
A simple approach would be to **always choose the country with the highest out-degree**, ensuring maximum follow-up options.

Why It Failed:

- **Doesn't account for opponent moves:** The opponent may also have many strong responses.
- **Leads to an evenly matched game:** No **long-term control** over the board.
- **May benefit the opponent:** High-out-degree nodes might give the opponent multiple options too.

Improvement: Out-Degree Differential Strategy

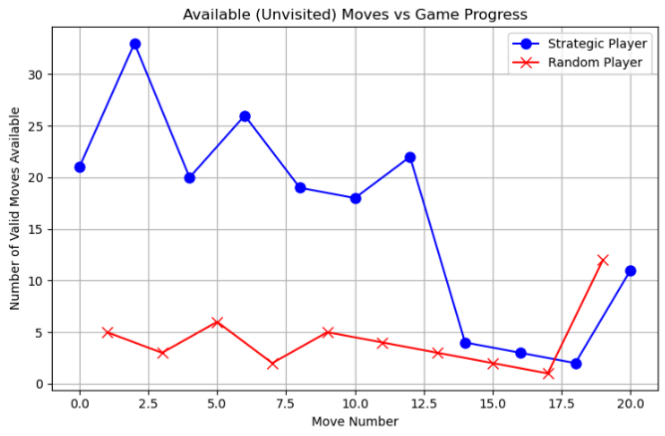
Instead of blindly maximizing out-degree, this strategy:

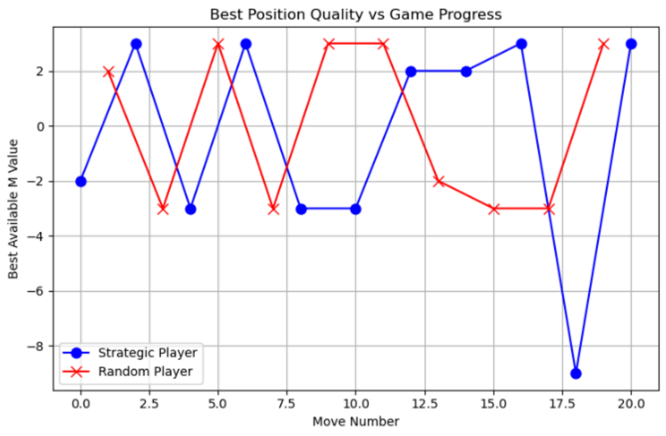
- **Minimizes the opponent's best possible out-degree**.
- Ensures that even if the opponent picks their best response, **we retain a move advantage**.

Simulation Results:

```
Both Strategic:
Player 1 Win Rate: 49.10%
Player 2 Win Rate: 50.90%

P1 Strategic vs P2 Random:
Player 1 Win Rate: 63.50%
Player 2 Win Rate: 36.50%

P1 Random vs P2 Strategic:
Player 1 Win Rate: 26.30%
Player 2 Win Rate: 73.70%

Both Random:
Player 1 Win Rate: 48.10%
Player 2 Win Rate: 51.90%
```

```
Starting position: abidjan
Move 0 (Strategic): ningbo - 21 moves available
Move 1 (Random): ottawa - 5 moves available
Move 2 (Strategic): adelaide - 33 moves available
Move 3 (Random): esfahan - 3 moves available
Move 4 (Strategic): nairobi - 20 moves available
Move 5 (Random): istanbul - 6 moves available
Move 6 (Strategic): lahore - 26 moves available
Move 7 (Random): edmonton - 2 moves available
Move 8 (Strategic): new taipei - 19 moves available
Move 9 (Random): ibadan - 5 moves available
Move 10 (Strategic): nnewi - 18 moves available
Move 11 (Random): islamabad - 4 moves available
Move 12 (Strategic): daegu - 22 moves available
Move 13 (Random): uyo - 3 moves available
Move 14 (Strategic): ouagadougou - 4 moves available
Move 15 (Random): urumqi - 2 moves available
Move 16 (Strategic): indore - 3 moves available
Move 17 (Random): ekurhuleni - 1 moves available
Move 18 (Strategic): izmir - 2 moves available
Move 19 (Random): raipur - 12 moves available
Move 20 (Strategic): rome - 11 moves available
Game Over at move 21 - no unvisited nodes available from rome
```





2. Entropy Minimization and Forced Predictability

> **Goal:** Make moves that keep your options open while forcing your opponent into **predictable** and **limited** future choices.

Failed Sub-Strategy: Just Minimizing Opponent's Out-Degree

Idea:
Always pick moves that leave the opponent with as few responses as possible.

Why It Failed:

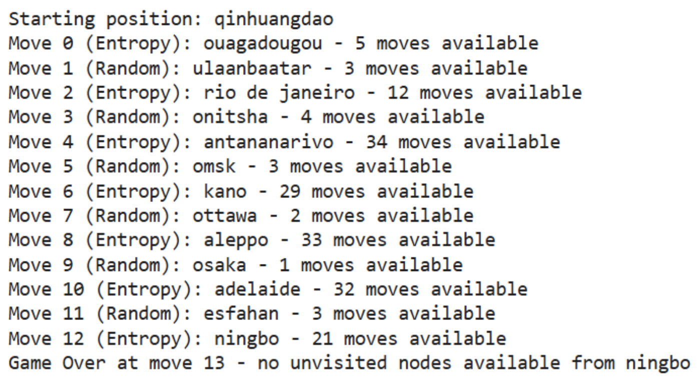
- **Some low-out-degree nodes lead to high-out-degree successors.**
- **Fails in early/mid-game**, when too many moves are available.
- **Doesn't account for move popularity** (players tend to pick familiar names).

Improvement: Shannon Entropy for Opponent Control

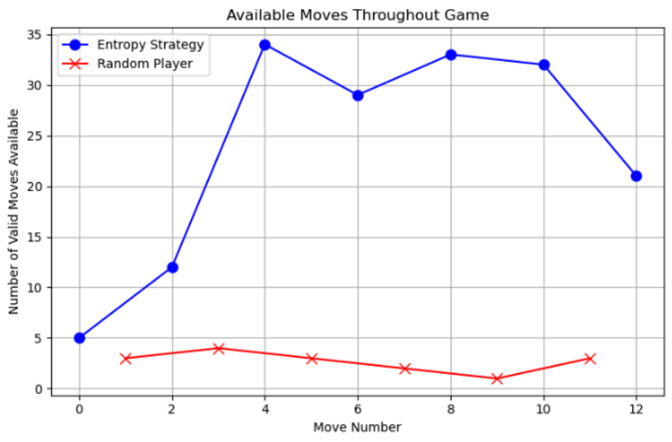
- Uses **Shannon entropy** to measure opponent unpredictability.
- Accounts for **popularity bias**, ensuring that even if the opponent has moves, **they are predictable**.

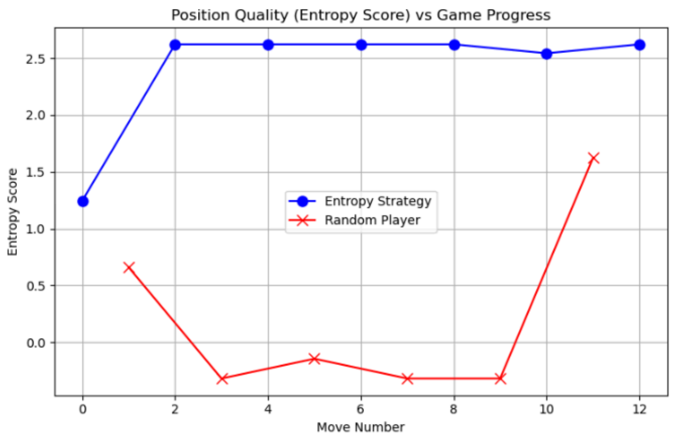
Simulation Results:

```
Both Entropy:
Player 1 Win Rate: 76.00%
Player 2 Win Rate: 24.00%

P1 Entropy vs P2 Random:
Player 1 Win Rate: 89.00%
Player 2 Win Rate: 11.00%

P1 Random vs P2 Entropy:
Player 1 Win Rate: 21.00%
Player 2 Win Rate: 79.00%
```



```
Starting position: qinhuangdao
Move 0 (Entropy): ouagadougou - 5 moves available
Move 1 (Random): ulaanbaatar - 3 moves available
Move 2 (Entropy): rio de janeiro - 12 moves available
Move 3 (Random): onitsha - 4 moves available
Move 4 (Entropy): antananarivo - 34 moves available
Move 5 (Random): omsk - 3 moves available
Move 6 (Entropy): kano - 29 moves available
Move 7 (Random): ottawa - 2 moves available
Move 8 (Entropy): aleppo - 33 moves available
Move 9 (Random): osaka - 1 moves available
Move 10 (Entropy): adelaide - 32 moves available
Move 11 (Random): esfahan - 3 moves available
Move 12 (Entropy): ningbo - 21 moves available
Game Over at move 13 - no unvisited nodes available from ningbo
```





3. Centralized Choke-Point Strategy

> **Goal:** Guide the opponent toward a **pre-determined weak position** by forcing them through a strategically chosen bottleneck node.

Failed Sub-Strategy: Just Picking High Betweenness Nodes

Idea:
Always move to a **high betweenness** node, assuming it will give us control.

Why It Failed:

- **Not all high-betweenness nodes are choke points.**
- **Fails to direct the opponent to a weak position.**
- **Doesn't consider game trajectory.**

Improvement: Choke-Point Frequency and Composite Score

- Identifies **specific choke nodes** that appear in **most paths** toward a weak position.
- Quantifies **choke effectiveness** using:
  - **Path frequency** (how often the node appears in critical paths).
  - **Betweenness centrality** (ensuring structural importance).
  - **Out-degree filtering** (to prevent self-traps).

---

4. Local Lookahead Strategy

> **Goal:** Simulate several turns in advance to find moves that give a **long-term advantage** while limiting the opponent's flexibility.

Failed Sub-Strategy: Greedy Move Maximization

Idea:
Always pick the move that maximizes your future options immediately.

Why It Failed:

- **Doesn't consider opponent responses.**

- **Fails in long-term strategy:** Some moves appear strong initially but lead to weak positions later.
- **Ignores network structure changes.**

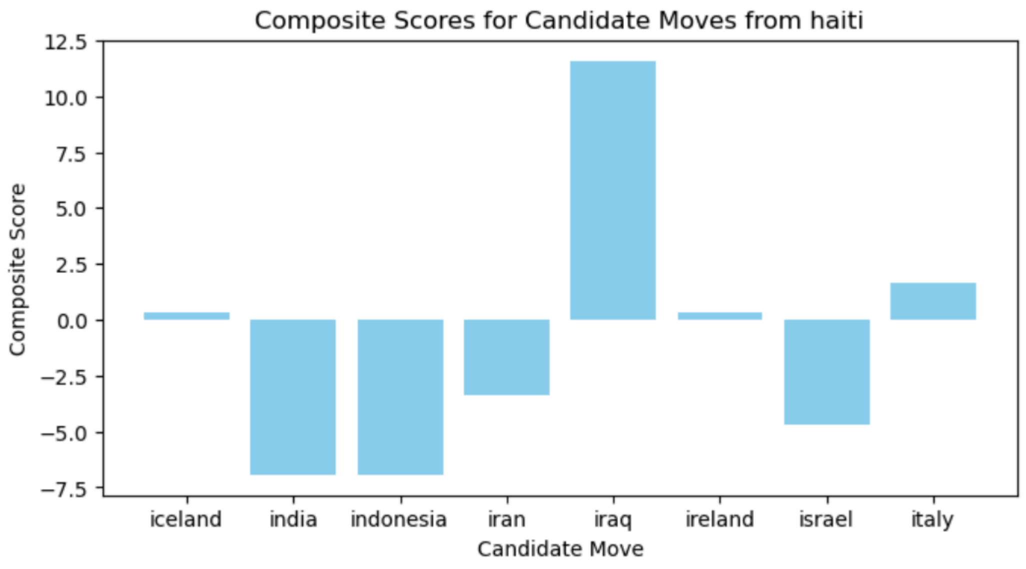
Improvement: Multi-Metric Lookahead

- **Simulates 2–3 turns ahead** to evaluate move impact.
- **Evaluates:**
    - **Opponent's Future Options**
    - **Our Future Options**
    - **Spectral Gap Changes** (preventing self-trapping).

Simulation Results:

```
P1 Lookahead vs P2 Random:
Player 1 Win Rate: 66.67%
Player 2 Win Rate: 33.33%

P1 Random vs P2 Lookahead:
Player 1 Win Rate: 33.33%
Player 2 Win Rate: 66.67%
```

```
Candidate: iceland, Score: 0.3017, Metrics: {'my_spec_gap': 0.6737109800556802, 'avg_opp_spec': 3.2, 'delta_spec': -2.52628
901994432, 'avg_my_moves': 10.944010695187165, 'avg_opp_moves': 9.4, 'delta_moves': 1.5440106951871648, 'H_self': 3.2679237
614023524, 'H_opp': 1.9839779803810733, 'delta_entropy': 1.283945781021279, 'composite_score': 0.3016674562641237}
Candidate: india, Score: -6.9200, Metrics: {'my_spec_gap': 0.8767876410218791, 'avg_opp_spec': 7.545454545454543, 'delta_sp
ec': -6.668666904432664, 'avg_my_moves': 11.097520661157025, 'avg_opp_moves': 11.181818181818182, 'delta_moves': -0.0842975
2066115626, 'H_self': 3.2914217751774415, 'H_opp': 3.458417206433648, 'delta_entropy': -0.1669954312562063, 'composite_scor
e': -6.919959856350026}
Candidate: indonesia, Score: -6.9200, Metrics: {'my_spec_gap': 0.8767876410218791, 'avg_opp_spec': 7.545454545454543, 'delt
a_spec': -6.668666904432664, 'avg_my_moves': 11.097520661157025, 'avg_opp_moves': 11.181818181818182, 'delta_moves': -0.084
29752066115626, 'H_self': 3.2914217751774415, 'H_opp': 3.458417206433648, 'delta_entropy': -0.1669954312562063, 'composite_
score': -6.919959856350026}
Candidate: iran, Score: -3.3591, Metrics: {'my_spec_gap': 0.8810141427021404, 'avg_opp_spec': 5.545454545454546, 'delta_spe
c': -4.664440402752406, 'avg_my_moves': 11.650032795487341, 'avg_opp_moves': 10.636363636363637, 'delta_moves': 1.013669159
1237046, 'H_self': 3.4627807864601214, 'H_opp': 3.1710760287075006, 'delta_entropy': 0.29170475775262084, 'composite_scor
e': -3.35906648587608}
Candidate: iraq, Score: 11.5850, Metrics: {'my_spec_gap': 3.0000000000000004, 'avg_opp_spec': 0.9999999999999998, 'delta_sp
ec': 2.000000000000001, 'avg_my_moves': 12.0, 'avg_opp_moves': 4.0, 'delta_moves': 8.0, 'H_self': 1.584962500721156, 'H_op
p': 0.0, 'delta_entropy': 1.584962500721156, 'composite_score': 11.584962500721156}
Candidate: ireland, Score: 0.3017, Metrics: {'my_spec_gap': 0.6737109800556802, 'avg_opp_spec': 3.2, 'delta_spec': -2.52628
901994432, 'avg_my_moves': 10.944010695187165, 'avg_opp_moves': 9.4, 'delta_moves': 1.5440106951871648, 'H_self': 3.2679237
614023524, 'H_opp': 1.9839779803810733, 'delta_entropy': 1.283945781021279, 'composite_score': 0.3016674562641237}
Candidate: israel, Score: -4.6916, Metrics: {'my_spec_gap': 1.4893305405599224, 'avg_opp_spec': 5.555555555555555, 'delta_s
pec': -4.066225014995632 5, 'avg_my_moves': 11.272727272727273, 'avg_opp_moves': 12.555555555555555, 'delta_moves': -1.28282
8282828282, 'H_self': 3.621462645940613, 'H_opp': 2.964019840191667, 'delta_entropy': 0.6574428057489463, 'composite_scor
e': -4.691610492074968}
Candidate: italy, Score: 1.6842, Metrics: {'my_spec_gap': 0.8767876410218791, 'avg_opp_spec': 0.9999999999999993, 'delta_sp
ec': -0.1232123589781202 2, 'avg_my_moves': 10.636363636363637, 'avg_opp_moves': 12.0, 'delta_moves': -1.3636363636363633,
'H_self': 3.1710760287075006, 'H_opp': 0.0, 'delta_entropy': 3.1710760287075006, 'composite_score': 1.684227306093017}

Best move from haiti: iraq with composite score 11.5850
```



Task 2 - Community Detection

Infomap

1. Modeling Game Flow as Information Flow

- **Random Walker Analogy**
  - Infomap treats the game as a flow of information by simulating a random walker that moves from one country to another following the atlas rule. For instance, if the walker

    - is at “India” (ending with “a”), it can only move to countries that begin with “A” (like “Australia”).
- **Direction Matters**
    - Since the atlas graph is directed, Infomap respects the game’s rule: moves only happen in the allowed direction (last letter → first letter). This ensures that the flow modeled by the algorithm exactly mimics the possible moves in the game.

---

2. The Map Equation and Community Detection

- **Minimizing Description Length** Infomap uses an information-theoretic concept called the map equation. It seeks to partition the graph into communities (clusters of countries) in such a way that the description length of a random walker’s journey is minimized. If the walker tends to stay within a certain group of countries, that group is identified as a community.
- **Two-Level Coding** The algorithm assigns codes at two levels:
    - **Module (Community) Level:** Each community gets a unique code.
    - **Node Level:** Within each community, each country is given its own code. When the walker moves within a community, only the node-level code is needed. However, moving between communities requires switching to a module-level code. A good community structure is one where most moves are internal, leading to an efficient (compressed) description.

---

1. **Interpreting the Computed Communities**

- **Natural Groupings:** In the atlas graph, communities detected by Infomap often represent clusters of countries that are closely interlinked by the game’s rules. For example:
    - **Letter Patterns:**
        - Countries might group together because many have names that start or end with similar letters. A community could consist mostly of countries that start with “A” or end with “n,” reflecting the natural bias in letter transitions.
    - **Choke Points:**
        - Some countries—like Yemen in our example—can act as bridges. Yemen, with incoming edges from countries ending in “y” and outgoing edges to those starting with “n,” might form a critical part of a community or even lie at the interface between communities. Such nodes have high betweenness and become strategic choke points.
- **Alignment with Human Intuition:**

  - When you look at the computed communities, you might notice that they correspond to groupings that humans would naturally recognize. People often group items based on common features; here, countries with similar starting or ending letters, or those that are strategically central in the flow of the game, naturally fall into the same clusters.

---

2. **Strategic Implications for the Game of Atlas**

- **Within-Community Moves:**
  - If you know that a group of countries forms a tightly connected community, playing within that community can be a safe strategy. It means you have many follow-up moves because the transitions (edges) are abundant within the community. This strategy minimizes the risk of leaving your opponent with an immediate winning move.
- **Exploiting Choke Points:** The Infomap algorithm might reveal that certain countries serve as bridges between communities. Choosing a country that lies on the border (or between) communities could force your opponent into a corner:
  - **Limiting Options:**
    - By steering the game through a choke point, you effectively funnel the game into a region where your opponent's valid moves are more restricted.
  - **Strategic Advantage:**
    - Even if you cannot immediately play the "ideal" country (for example, you may want to say "Hungary" but the current flow does not allow it), you can opt for a move that guides the game toward that region. This long-term planning—based on the community structure—gives you a competitive edge.
- **Intuition of a "Good" Strategy:**
  - The communities provide insight into the structure of valid moves:
  - **Control the Flow:**
    - By understanding which countries are central within a community or connect multiple communities, you can prioritize moves that maximize your control over the game's progression.
  - **Predictability:**
    - A move within a dense community might give you more predictable follow-up moves, while a move that transitions between communities could be used to disrupt your opponent's expected path.

GAE with GCN and HDBSCAN

1. What is GAE (Graph Autoencoder)?

A **Graph Autoencoder (GAE)** is a **neural network model** that learns **low-dimensional representations (embeddings)** of nodes in a graph while preserving structural relationships.

- **Encoder:** Compresses node features into a **latent space**.
- **Decoder:** Reconstructs edges from embeddings, ensuring learned representations retain graph structure.
- **Loss Function:** Minimizes reconstruction error, ensuring embeddings capture key node relationships.

---

2. Using GCN (Graph Convolutional Networks) as the Encoder

A **GCN (Graph Convolutional Network)** is a **type of neural network designed for graphs**, which learns node representations by aggregating information from neighbors.

In **GAE for the Game of Atlas**:

- **Nodes:** Countries
- **Edges:** Valid game transitions (last-letter to first-letter rule).
- **Node Features:** A **52-dimensional one-hot encoding** (first and last letter encoded separately).

GCN Encoder Layers:

1. **First Layer:** Aggregates neighboring features to learn local structure.
2. **Activation (ReLU):** Introduces non-linearity.
3. **Second Layer:** Maps to a **latent space (4D vector per country)**.

---

3. HDBSCAN: Clustering the Learned Embeddings

Once the **GCN-based encoder** learns **low-dimensional embeddings** for each country, we apply **HDBSCAN (Hierarchical Density-Based Clustering)** to group similar countries.
