Visualizing and Analyzing Popular Sports Communities and Connections using Youtube Data

Ethan Hsu, Kaden Kram

Advisor: Professor Rebrova

Submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Science in Engineering

Department of Operations Research and Financial Engineering

Princeton University

April 2025

I hereby declare that I am the sole author of this project.

I authorize Princeton University to lend this project to other institutions or individuals for the purpose of scholarly research.

_____

Ethan Hsu, Kaden Kram

I further authorize Princeton University to reproduce this project by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

_____

Ethan Hsu, Kaden Kram

Abstract

YouTube serves as a central platform for modern sports consumption, yet the intricate connections and overlaps between large channel communities remain largely underexplored. Traditional audience analysis often relies on declared preferences, missing rich behavioral data embedded in user engagement. This research addresses this gap by mapping and analyzing the interconnection network of prominent YouTube sports channels based on shared audience activity. We collected commenter data from 95 major sports channels (sampling up to 20,000 unique commenters per channel) and constructed a network where connections represent significant commenter overlap ($\geq$ 100 shared). Employing network visualization (ForceAtlas2, YiFan Hu), community detection (Louvain Modularity), and quantitative analysis (Scaled Betweenness Centrality, Scaled Eigenvector Centrality, Clustering Coefficient), we uncover the underlying structure of this digital sports ecosystem.

Our findings reveal distinct community clusters primarily aligned with European Football/Global Soccer, US Sports/Highlight Culture, Fitness/Extreme Sports, Brazilian Soccer, and Cricket, alongside numerous isolated channels. We identify key "bridge" channels, such as major organizations (FIFA, Premier League) and broadcasters (ESPN, TNT Sports Brasil), that facilitate connections between these communities. Influential hubs are concentrated within specific clusters, particularly major European football clubs (high Eigenvector Centrality), while clustering coefficients quantify the varied local cohesion across different sports niches. This study provides a data-driven map of audience interaction patterns, offering valuable insights for content creators, marketers, and sports organizations seeking to understand and navigate the complex landscape of digital sports fandom. By leveraging behavioral data, our approach reveals nuanced community structures and cross-sport engagement dynamics often invisible to traditional methods.

Contents

List of Figures

Chapter 1

Introduction

YouTube has fundamentally transformed how sports content is consumed and how fan communities interact. With over 2.5 billion monthly active users, it serves as the world's largest video platform where sports fans gather to watch highlights, reactions, and other sports content. Unlike traditional media consumption, YouTube creates unique two-way engagement opportunities through subscriptions and comments, fostering a visible community that can be analyzed through YouTube's API.

Sports fandom has historically been studied through traditional methods like surveys, viewership metrics, and declared preferences. These approaches often treat sports communities as isolated entities with minimal overlap. However, digital platforms have created communities that have never before been seen and created new patterns of engagement that have never been researched or visualized before.

1.1 Establishing the Importance

Understanding the interconnectedness of sports communities on Youtube addresses several important gaps:

Theoretical Significance: Network Theory applied to online sports communities could reveal properties that conventional research misses. The patterns of shared

viewership and engagement across different sports represent a new approach to understanding fandoms in this new digital age.

Practical applications: Content creators, sports marketers, and media organizations can leverage these insights to develop strategies to expand their audiences. Identifying bridge channels and community clusters allows for more targeted content creation and community-building initiatives.

Cultural relevance: As sports consumption increasingly moves online, understanding how digital communities form around sports content becomes crucial for comprehending modern fan behavior. Traditional fandoms or boundaries between sports may be dissolving in digital spaces, creating new hybrid communities.

New Methodology: By analyzing comment data at scale, you get objective, behavior-based information, and you get more authentic community structures when compared to the older self-reported survey-based approaches.

By mapping the network of shared commenters across major sports channels, this research provides invaluable insights for how different sport communities on YouTube interact, connect, and overlap which is information largely unexplored despite its significance for understanding contemporary sports culture.

1.2 Research Objectives and Questions

This project is guided by the following research questions:

1. What distinct community clusters emerge when analyzing shared commenters across major sports channels on Youtube?

2. Which YouTube sports channels serve as "bridges" between different sports communities, and what characteristics do these bridge channels share?

3. To what extent do traditionally separate sports fandoms (like soccer and basketball for example) overlap in their digital engagement patterns?

4. How can network analysis reveal hidden patterns in the connections between sports communities

To address these questions, our project applies network analysis principles to map the complex ecosystem of sports communities on YouTube. Specifically, we aim to:

- Identify different clusters of sports channels using modularity on shared commenter activity
- Use an efficient and high-quality layout algorithm to help us visualize the network, specifically helping us discover "bridge channels" that connect multiple sports communities.
- Analyze unexpected fandom connections that challenge conventional understanding of sports audience segmentation
- Quantify the strength of connections between different sports communities.

1.3 Structure of the Project Write Up

The remainder of the project write up is organized as follows:

- In Chapter 2, we will go over our YouTube data collection and cleaning process which will be used to create our network.
- In Chapter 3, we will detail our methodology, from the construction of our network based on shared commenters, to the analytical techniques employed to find clusters and patterns.

- Chapter 4 presents our findings, showcasing the network visualization, identified community clusters, bridge channels, and the strength of connections between different sports communities.

- In Chapter 5, we discuss the implications of our findings for content creators, marketers, and sports organizations, as well as the theoretical contributions to understanding modern digital sports communities.

- Finally, Chapter 6 concludes this project by summarizing key insights, acknowledging limitations, and suggesting directions for future research.

With this structure, we aim to provide a comprehensive analysis of how sports fandoms interact and overlap in the digital space, offering valuable insights for both academic understanding and practical applications in the sports media ecosystem.

Chapter 2

Data Collection Methodology

This project required us to employ a structured data collection approach to analyze the relationships among the top YouTube sports channels. This collection process uses a mixed-method approach that combines automated web scraping techniques with manual verification to ensure quality data. The primary data source was Social Blade (https://socialblade.com), a recognized analytics platform that tracks and ranks YouTube channel statistics. The final dataset comprises 95 nodes (channels), representing 95% of the initially targeted sample of prominent sports content creators. We then used the YouTube API to find 20000 unique commentor IDs for each channel, helping us create weighted edges that represent the number of shared commentors between users.

2.1 Gathering the Channels

Channel names and subscriber counts were extracted from Social Blade's sports category rankings through a hybrid collection approach utilizing both automated scraping techniques and manual verification. The primary reference URL [3] became broken (or deleted) during the study, which required me to supplement the rest of the dataset with an alternative Social Blade categorization [4]. This alternative list only exhibited

partial differences from the original list.

2.2 Channel ID Extraction

This hybrid collection process had an initial custom scraping algorithm phase utilizing a package called BeautifulSoup4 that can read the HTML data of a website. Using this package, I was able to extract the channel name and the subscriber counts. Following the initial data collection, I used the YouTube API to extract unique channel IDs for each target channel. These IDs served as an identifier necessary for further API interactions as there are multiple YouTube channels with the same name, and we needed a way to differentiate them if we had similarly named nodes.

```python
youtube = build(YOUTUBE_API_SERVICE_NAME, YOUTUBE_API_VERSION, developerKey=API_KEY)

def get_channel_id(channel_name):
    """
    Search for a YouTube channel by name and return its ID.
    """
    try:
        # Search for the exact channel name
        search_response = youtube.search().list(
            q=channel_name,
            part="id,snippet",
            maxResults=5,
            type="channel"
        ).execute()

        # Check if we got any results
        if not search_response["items"]:
            print(f"No results found for: {channel_name}")
            return None

        # Look for an exact match first
        for item in search_response["items"]:
            if item["snippet"]["title"].lower() == channel_name.lower():
                channel_id = item["id"]["channelId"]
                print(f"Found exact match for {channel_name}: {channel_id}")
                return channel_id

        # If no exact match, return the first result
        channel_id = search_response["items"][0]["id"]["channelId"]
        actual_name = search_response["items"][0]["snippet"]["title"]
        print(f"Using closest match for {channel_name}: {actual_name} (ID: {channel_id})")
        return channel_id

    except Exception as e:
        print(f"Error searching for {channel_name}: {e}")
        return None
```

Figure 2.1: Function for searching for a YouTube Channel's ID through its name

```python
def verify_channel_id(channel_id):
    """
    Verify that a channel ID exists and get its actual title and subscriber count.
    """
    try:
        channel_response = youtube.channels().list(
            part="snippet,statistics",
            id=channel_id
        ).execute()

        if not channel_response["items"]:
            return None, None

        return (
            channel_response["items"][0]["snippet"]["title"],
            channel_response["items"][0]["statistics"].get("subscriberCount", "0")
        )
    except Exception as e:
        print(f"Error verifying channel ID {channel_id}: {e}")
        return None, None
```

Figure 2.2: Function for verification of a channel ID and getting the actual title of the channel and its subscriber count

After running these functions, I stored the data in a master spreadsheet to review the differences between the Social Blade web scrape and the channel ID extraction and manually fix the discrepancies.

Channel ID	Channel Name	Actual Channel Name	Subscriber Count (CSV)	Subscriber Count (API)
UCJ5v_MCY6G	WWE	WWE	107M	107000000
UCja8sZ2T4yIlc	Sports	CBS Sports	74.7M	985000
UCtxD0x6AuNN	UR - Cristiano	UR - Cristiano	74.1M	74100000
UCRijo3ddMTh	Dude Perfect	Dude Perfect	60.8M	60800000
UCOnIJiQuk1fE	YOLO AVENTUI	YOLO AVENTURAS	60.3M	60300000
UC1a2ZCw7tuç	Celine Dept	Celine Dept	44.3M	44300000
UCWsDFcIhY2I	IShowSpeed	IShowSpeed	36.4M	36600000
UCQIUhhcmXs	Jesser	Jesser	25.8M	26200000
UC5f5IV0Bf79Y	How Ridiculous	How Ridiculous	23.5M	23500000
UCpcTrCXblq7ł	FIFA	FIFA	23M	23000000
UCWJ2lWNubA	NBA	NBA	22.8M	22800000
UCblfuW_4rakI	Red Bull	Red Bull	22M	22100000

Figure 2.3: Snippet of final channel ID spreadsheet with subscriber counts

2.2.1 Technical Constraints

Sample Coverage: This process achieved a 95% coverage rate ($\frac{95}{100}$ target channels). This represents a statistically robust sample despite minor data collection impediments. We think that these data accessibility barriers could be due to API limits or YouTube Geographic restrictions.

2.3 Collecting the Commenters

```python
def get_uploads_id_for_channel(youtube, channel_id):
    try:
        response = youtube.channels().list(
            part='contentDetails,statistics',
            id=channel_id
        ).execute()

        if 'items' in response and response['items']:
            item = response['items'][0]
            # Print the structure for debugging
            print(f"Channel response keys: {item.keys()}")

            if 'statistics' in item and 'videoCount' in item['statistics']:
                video_count = int(item['statistics']['videoCount'])
                print(f"Video count: {video_count}")

                if video_count > 0 and 'contentDetails' in item:
                    if 'relatedPlaylists' in item['contentDetails']:
                        if 'uploads' in item['contentDetails']['relatedPlaylists']:
                            uploads_id = item['contentDetails']['relatedPlaylists']['uploads']
                            return uploads_id

        print(f'Could not find uploads id for channel id: {channel_id}')
        return None
    except Exception as e:
        print(f"Error getting uploads ID: {e}")
        # Print the full response for debugging
        print(f"Response: {response}")
        return None
```

Figure 2.4: Function that gets a channels playlist of most recently uploaded video IDs

```python
def get_commenters_for_uploads_id(youtube, uploads_id, max_commenters_per_channel):
    channel_commenters = set()
    video_next_page_token = ""

    while True:
        if video_next_page_token is None:
            print(f"Found {len(channel_commenters)} commenters")
            return channel_commenters

        video_ids, video_next_page_token = get_videos_for_uploads_id(youtube, uploads_id, video_next_page_token)

        if len(video_ids) == 0:
            print(f'Empty videos list indicates playlistID was invalid')
            return None

        comment_disabled_video_count = 0
        for video_id in video_ids:
            video_commenters = get_commenters_for_video(youtube, video_id, channel_commenters, max_commenters_per_channel)

            if video_commenters is None:
                comment_disabled_video_count += 1
                if comment_disabled_video_count > 10:
                    print(f'Channel exceeded 10 comment disabled videos')
                    return None
                continue

            if len(channel_commenters.union(video_commenters)) > max_commenters_per_channel:
                unseen_commenters = set()
                for commenter in video_commenters:
                    if commenter not in channel_commenters:
                        unseen_commenters.add(commenter)
                        if len(unseen_commenters) + len(channel_commenters) >= max_commenters_per_channel:
                            channel_commenters.update(unseen_commenters)
                            print(f'Successfully reached {len(channel_commenters)} commenters')
                            return channel_commenters
            else:
                channel_commenters.update(video_commenters)
```

Figure 2.5: Function that collects commenters for a channel's upload ID playlist

Using these functions, we collected a total of 20000 unique commenters from each of the 95 channels that we scraped. More specifically, we saved each channel's commenters in a pickle file to later help us create an edge csv file that we can use later for network creation.

2.3.1 Exception Handling and Data Remediation

Resolved Technical Exceptions: The channel Colin Amazing had comments disabled, so we had to substitute it with a comparable channel (found from the new Social Blade link). Same with "Sports", the code replaced it with "CBS Sports".

Pending Data Quality Issues: Arabic channel names had to be sanitized for file saving. Vikram Singh Fitness exhibited patterns consistent with artificial engagement, making it hard for me to reach the 20000 unique commenter number.

2.4 Graph Construction Preparation

After having all the pickle files, we created edges between the channels if they shared at least 100 commenters between both of their 20000 commenters. We chose 100 because real-world networks are sparse, and we wanted more edges between nodes to analyze clusters, but also a high enough number not to consider bot comments. The edges file had 3 columns: source, target, and weight (number of shared commenters).

```python
# Calculate overlaps
edges = []
threshold = 100  # Minimum shared commenters threshold

for channel1 in all_commenters:
    for channel2 in all_commenters:
        if channel1 >= channel2:  # Avoid duplicates and self-comparisons
            continue

        shared_commenters = all_commenters[channel1].intersection(all_commenters[channel2])
        shared_count = len(shared_commenters)

        if shared_count >= threshold:
            edges.append({
                "Source": channel1,
                "Target": channel2,
                "Weight": shared_count
            })

# Create edges CSV
edges_df = pd.DataFrame(edges)
edges_df.to_csv("sports_channel_edges.csv", index=False)
print(f"Generated {len(edges)} edges between channels")
files.download("sports_channel_edges.csv")
```

Figure 2.6: Code for creating the edges files by calculating overlapping commenters

Source	Target	Weight
FLA TV	FutParódias	289
FLA TV	TNT Sports Bras	360
CazéTV	FLA TV	289
CazéTV	ESPN Brasil	1923
CazéTV	Vosso Canal	157
CazéTV	Renato Cariani	181
CazéTV	Desimpedidos	335
CazéTV	FutParódias	386
CazéTV	TNT Sports Bras	1637
Alex Segura LR	Browney en Esp	151

Figure 2.7: Snippet of edges file for network construction

Similarly, we created a node file that contained the name of the channel along with the number of subscribers the channel had. This way, I can size the nodes based on the channel's number of subscribers.

```
# Create nodes dataframe
nodes = []
for idx, row in channels_df.iterrows():
    # Only include channels that we collected commenters for
    if row['Channel Name'] in all_commenters:
        nodes.append({
            "Id": row['Channel Name'],  # Using name as ID for simplicity
            "Label": row['Actual Channel Name'],
            "Subscribers": row['Subscriber Count (API)'],
            "CommentersCount": len(all_commenters[row['Channel Name']])
        })

nodes_df = pd.DataFrame(nodes)
nodes_df.to_csv("sports_channel_nodes.csv", index=False)
print(f"Generated nodes file with {len(nodes)} channels")
files.download("sports_channel_nodes.csv")
```

Figure 2.8: Node File creation code after you upload the master channel id sheet

UR - Cristiano	UR - Cristiano	74100000	20000
Dude Perfect	Dude Perfect	60800000	20000
YOLO AVENTUI	YOLO AVENTUI	60300000	20000
Celine Dept	Celine Dept	44300000	20000
IShowSpeed	IShowSpeed	36600000	20000
Jesser	Jesser	26200000	20000
How Ridiculous	How Ridiculous	23500000	20000
FIFA	FIFA	23000000	20000
NBA	NBA	22800000	20000
Red Bull	Red Bull	22100000	20000

Figure 2.9: Snippet of Node File for graph construction

Chapter 3

Network Construction and Analysis Methodology

This chapter details the methodological approach employed to construct and analyze the network of YouTube sports channels based on shared commenter activity. After the data collection and cleaning processes described in Chapter 2, we used a systematic approach to visualize and analyze the relationships between these channels. In Chapter 4, we will show the visualizations that came from this methodology and talk about the interpretation of our results.

3.1 Network Layout Construction

3.1.1 Data Import and Initial Setup

After obtaining our edges and nodes files from our data collection process, we imported this data into Gephi, an open-source network visualization and analysis software. The nodes file contained information about each YouTube channel, including its name and subscriber count, while the edges file represented connections between channels based on shared commenters, with edge weights proportional to the number

of common commenters.

The initial import yielded a network consisting of 95 nodes (channels) connected by weighted edges, where edge weights represented the number of shared commenters between two channels (with a minimum threshold of 100 shared commenters as established in Section 2.4).

To effectively visualize the network structure, we employed two force-directed layout algorithms, specifically adapted to handle our weighted network where edge weights represent the number of shared commenters between channels. This way we had two different network layouts to compare and analyze.

3.1.2 ForceAtlas2 Algorithm

For our first visualization, we employed ForceAtlas2 [2], a force-directed layout algorithm specifically designed for network visualization. This algorithm simulates a physical system to arrange nodes in a way that reveals the underlying structure of our YouTube sports channel network.

The algorithm operates on the principle of simulating physical forces between nodes:

- Nodes repel each other like charged particles
- Edges act like springs that pull connected nodes together
- A central gravity force prevents disconnected components from drifting apart

Mathematically, for our weighted network $G = (V, E, W)$ where nodes represent YouTube channels, edges represent connections, and weights represent shared commenters, the force model consists of:

$$F_r(i, j) = k_r \frac{(deg(i) + 1)(deg(j) + 1)}{d_{ij}} \tag{3.1}$$

Where F_r is the repulsive force between nodes i and j, $deg(i)$ is the degree of node i, d_{ij} is the Euclidian distance ($\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$) between nodes i and j, and k_r is the repulsion constant controlled by the scaling parameter.

$$F_a(i,j) = d_{ij} \times w_{ij}^{\alpha} \tag{3.2}$$

Where F_a is the attractive force, w_{ij} is the edge weight (number of shared commenters), and α is the edge weight influence parameter (set to 1.0 in our implementation).

$$F_g(i) = k_g \times (deg(i) + 1) \tag{3.3}$$

Where F_g is the gravity force pulling node i toward the center, and k_g is the gravity constant.

With LinLog mode enabled, the repulsive force formula is modified to:

$$F_r(i,j) = k_r \frac{(deg(i)+1)(deg(j)+1)}{log(1+d_{ij})} \tag{3.4}$$

This modification emphasizes community structures by making the repulsive force decrease more slowly with distance.

The total force acting on each node is the vector sum of all forces:

$$F_{total}(i) = \sum_{j \neq i} F_r(i,j) + \sum_{j \in N(i)} F_a(i,j) + F_g(i) \tag{3.5}$$

Where $N(i)$ is the set of nodes connected to node i.

Once the total force for each node is calculated, the node positions are updated according to:

$$x_i^{(t+1)} = x_i^{(t)} + \delta x_i \tag{3.6}$$

$$y_i^{(t+1)} = y_i^{(t)} + \delta y_i \tag{3.7}$$

Where the displacement $(\delta x_i, \delta y_i)$ is controlled by the scaling factor and a "Tolerance (speed)" parameter that regulates oscillation in the system:

$$\delta x_i = F_{total,i,x} \times s \times \tau \tag{3.8}$$

$$\delta y_i = F_{total,i,y} \times s \times \tau \tag{3.9}$$

Here, $(x_i^{(t)}, y_i^{(t)})$ is the position of node i at iteration t, s is the scaling factor (set to 2.0 which I will mention later), and τ represents the "Tolerance (speed)" parameter that controls how much oscillation is allowed during layout (which is set to 1).
For our implementation analyzing YouTube sports communities, ForceAtlas2 was configured with the following parameters:

- **Scaling**: 2.0 - Controls the strength of repulsion between nodes
- **Gravity**: 1.0 - Determines how strongly nodes are pulled toward the center
- **Edge Weight Influence**: 1.0 - Full impact of edge weights on attractive forces
- **LinLog mode**: Enabled - Better represents community structures
- **Prevent Overlap**: Enabled, will be explained in Section 3.1.4 the reason.
- **Tolerance (speed)**: 1.0 - Default value to help with precision and reduce oscillations.

The algorithm proceeds through an iterative process:

1. Initialize node positions randomly
2. Calculate repulsive forces between all pairs of nodes

3. Calculate attractive forces between connected nodes

4. Calculate gravity forces pulling nodes toward the center

5. Update node positions based on the sum of forces

6. Repeat steps 2-5 until the system stabilizes

This configuration ensures that YouTube channels sharing many commenters (higher edge weights) are positioned closer together in the visualization. The gravitational parameter keeps isolated nodes from drifting too far while still allowing natural clustering to emerge. The LinLog mode enhances community detection, which is particularly important for our analysis of distinct sports channel clusters.

The resulting layout effectively reveals:

- Community structures of sports channels based on audience overlap
- Bridge channels positioned at boundaries between communities
- The relative strength of audience connections through spatial proximity

3.1.3 YiFan Hu's Proportional Algorithm

For our second visualization, we employed YiFan Hu's Proportional Algorithm [1] to generate an independent layout of the same network data. This allowed us to compare how different algorithms interpret the relationships between YouTube sports channels based on shared commenters.

The algorithm operates on the principle of simulating physical forces between nodes:

- Nodes repulse each other like electrical charges
- Edges act like springs connecting nodes

- The algorithm aims to minimize the energy of this physical system

Mathematically, the system seeks to minimize a global energy function $E(X)$:

$$E(X) = \sum_{i<j} E_{rep}(i,j) + \sum_{(i,j)\in E} E_{att}(i,j) \tag{3.1}$$

Where repulsive energy follows an inverse-power law:

$$E_{rep}(i,j) = \frac{(deg(i)+1)(deg(j)+1)}{d_{ij}^2} \tag{3.2}$$

And attractive energy is proportional to edge weight (shared commenters):

$$E_{att}(i,j) = -w_{ij} \times k_{opt} \times \log\left(\frac{d_{ij}}{k_{opt}}\right) \tag{3.3}$$

We can use these forces to update and make tentative movements to our nodes:

$$x_i^{(t+1)} = x_i^{(t)} + F_{\text{total},i,x} \times \Delta t \tag{3.4}$$

$$y_i^{(t+1)} = y_i^{(t)} + F_{\text{total},i,y} \times \Delta t \tag{3.5}$$

where Δt is the current step length

We can then see if these movements will increase/decrease the overall system energy. Looking at equation 3.7, we can adaptively adjust the step length:

- Increases step length (but never beyond the maximum step size) if energy decreased
- Decreases step length using the step ratio (0.95 in our implementation) if energy increased

This repeats until either the energy improvement falls below a predefined convergence threshold (indicating stabilization) or the maximum number of iterations is reached (typically 100 iterations in Gephi's default implementation).

What makes YiFan Hu's approach distinctive is its computational efficiency, achieved through:

1. **Multilevel technique**: The algorithm generates a sequence of progressively smaller graphs from the original, each capturing essential connectivity. It then applies the force-directed algorithm to this sequence from small to large, using each smaller graph's layout as the starting point for the larger one. This process:

   - Creates a hierarchy of simplified graphs $G_0, G_1, ..., G_k$
   - Computes optimal layouts starting with the simplest graph
   - Progressively refines through each level to the full network
   - Helps avoid local minima and reveals hierarchical community structures

2. **Barnes-Hut approximation**: This reduces computational complexity from $O(n^2)$ to $O(n \log n)$ by approximating the repulsive forces from clusters of distant nodes as if they were a single "super-node":

$$F_{Barnes-Hut}(i, C) = \begin{cases} \sum_{j \in C} F_r(i, j) & \text{if } \frac{s}{d_{i,c_m}} < \theta \\ |C| \times F_r(i, c_m) & \text{otherwise} \end{cases} \tag{3.6}$$

   where d_{i,c_m} is the distance between node i and the center of mass c_m of cluster C, S is the spatial size (e.g., diameter) of cluster C, and θ is a resolution parameter.

3. **Adaptive step length**: The algorithm dynamically adjusts how far nodes move in each iteration based on whether the previous move decreased the system's energy:

$$\Delta x_i^{(t+1)} = \begin{cases} \min((1+\beta) \times \Delta x_i^{(t)}, \Delta_{max}) & \text{if } \Delta E < 0 \\ \frac{\Delta x_i^{(t)}}{(1+\gamma)} & \text{otherwise} \end{cases} \tag{3.7}$$

Implementation Details

For our YouTube sports channel network, we configured the algorithm with the following parameters:

- **Optimal Distance**: 60 - The natural equilibrium distance between connected nodes with unit weight. We chose it because it looked best when running the layout algorithm, the disconnected components, and clusters are not too far apart.
- **Relative Strength**: 0.2 - Controls the balance between attractive and repulsive forces
- **Initial Step Size**: 20 - Maximum initial movement distance per iteration
- **Step Ratio**: 0.95 - Controls the rate of adaptive step size adjustment
- **Adaptive Cooling**: Enabled - Dynamically adjusts convergence parameters

The algorithm follows this general procedure:

1. Initialize node positions randomly
2. Set an initial step length (20 in our case)
3. For each iteration:
    - Calculate attractive forces between connected nodes, weighted by shared commenters
    - Calculate repulsive forces between all node pairs (using Barnes-Hut approximation)
    - Update node positions based on these forces
    - Adaptively adjust the step length based on energy changes

4. Continue until the layout stabilizes (energy improvement falls below threshold)

Comparing this independent visualization with our ForceAtlas2 layout provides valuable insights: consistent findings across both algorithms strengthen our conclusions about community structures, while differences highlight areas of ambiguity in the network structure.

3.1.4 Post-Processing Adjustments

After the main layout algorithms, we applied three additional algorithms (as well as manually moving extremely far away distant nodes) to enhance readability:

1. **Noverlap Algorithm**: Prevents node overlap through a force-directed approach: $F_{\text{overlap}}(i,j) = \begin{cases} k_{\text{overlap}} \times (r_i + r_j - d_{ij}) \times \boldsymbol{d_{ij}} & \text{if } d_{ij} < (r_i + r_j) \\ 0 & \text{otherwise} \end{cases}$

   Where:

   - r_i and r_j are the radii of nodes i and j (proportional to subscriber count in our case)
   - $\boldsymbol{d_{ij}}$ is the unit vector from node i to node j
   - k_{overlap} is a constant controlling the strength of the overlap force

2. **Contraction Algorithm**: Improves overall compactness by moving nodes toward cluster centers: $x_i' = x_i + k_{\text{contraction}} \times (c_x - x_i)$ $y_i' = y_i + k_{\text{contraction}} \times (c_y - y_i)$ Where (c_x, c_y) are the coordinates of the center of mass of the node's cluster and $k_{\text{contraction}}$ is the contraction factor (set to 0.8).

3. **Label Adjust Algorithm:** Enhances label readability by repositioning labels to avoid overlap while maintaining proximity to their nodes. The algorithm

works by simulating physical forces among labels:

$$F_{repel}(i,j) = \begin{cases} k_{repel} \times \frac{(w_i+w_j)\times(h_i+h_j)}{d_{ij}^2} \times \mathbf{d_{ij}} & \text{if labels overlap} \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

Where w_i and h_i are the width and height of label i, and $\mathbf{d_{ij}}$ is the unit vector pointing from the center of label j to the center of the label i.

This multi-step layout process allowed us to effectively reveal the underlying structure of YouTube sports channels while maintaining the readability of the network visualization. The mathematical foundations of the layout algorithms is to ensure that the final layout accurately represents the weighted relationships between channels, with edge weights (shared commenters) directly influencing the spatial arrangement of nodes. We hope to see that these layouts will be successful with aligning with community detection when looking for clusters.

3.2 Community Detection

To identify distinct communities within the network, we employed the Louvain method for community detection. This algorithm optimizes modularity, which measures the strength of the division of a network into communities. Modularity is based on comparing the density of connections inside communities with the density of connections between communities.

3.2.1 Louvain Modularity Algorithm

The Louvain method operates on the principle of modularity maximization, where modularity (Q) quantifies the quality of community assignments by measuring the density of links within communities compared to links between communities:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \tag{3.9}$$

Where:

- A_{ij} represents the weight of the edge between nodes i and j (shared commenters in our case)
- k_i and k_j are the sum of weights of edges attached to nodes i and j respectively
- m is the sum of all edge weights in the network
- $\delta(c_i, c_j)$ equals 1 if nodes i and j belong to the same cluster, and 0 otherwise

The algorithm proceeds through an iterative two-phase process:

1. **Local Optimization Phase**: Each node is initially assigned to its own community. Then, for each node i, the algorithm evaluates the modularity gain achieved by moving i from its current community to each of its neighboring communities. The node is placed in the community that yields the maximum positive gain in modularity. This process continues until no further improvement can be achieved.
2. **Network Aggregation Phase**: Once the first phase is complete, a new network is constructed where nodes represent the communities found in phase one. The weights of edges between these new nodes are calculated as the sum of weights of edges between nodes in the corresponding communities. Self-loops represent edges within communities.

These two phases are repeated iteratively until no further modularity improvement is possible, resulting in a hierarchical decomposition of the network.

For our initial implementation analyzing YouTube sports communities, the Louvian algorithm will be configured to the following parameters. However, we will compare different parameters accordingly after our visualizations if different clusters give us different interpretations.

- **Resolution**: 1.0 - Controls the granularity of the detected communities (higher values lead to smaller communities)
- **Randomization**: On - Addresses potential sensitivity to node order
- **Use edge weights**: Enabled - Accounts for the strength of connections (number of shared commenters)
- **Randomization**: Enabled - Ensures reproducibility of results

We hope that the detected communities provide valuable insights into how sports audiences on YouTube naturally cluster based on their commenting behavior, revealing patterns that may not be apparent when analyzing declared preferences or traditional demographic data.

3.3 Network Analysis Methodology and Metrics

To identify important channels within the network structure, we look at three important network measurements. We hope that these metrics would give us a better understanding of the sports channel ecosystem beyond just identifying communities, but also allowing us to characterize the structure and dynamic of different channels and communities on YouTube.

3.3.1 Betweenness Centrality

Betweenness centrality measures the extent to which a node serves as a bridge along the shortest paths between other nodes in the network. For a channel v in our

YouTube sports network, betweenness centrality $C_B(v)$ is defined as:

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{3.10}$$

Where σ_{st} represents the total number of shortest paths from channel s to channel t, and $\sigma_{st}(v)$ denotes the number of those paths passing through channel v. We normalize this value by dividing by $(n-1)(n-2)/2$, where n is the number of channels in our network.

In our analysis, channels with high betweenness centrality function as critical connectors between different sports communities, potentially reaching audiences across traditional sport boundaries. These bridge channels likely feature content with cross-sport appeal, such as general sports news, fitness training applicable to multiple sports, or personality-driven content attracting diverse fan bases. By identifying these channels, we can understand the pathways through which different sports audiences interact and overlap within the broader YouTube ecosystem.

To account for the node weight, we will use a logarithmic correcting factor where each centrality score will be multiplied by:

$$1 + \log_{10}\left(\frac{\text{Most Subscribers of any Node (WWE)}}{\text{Current Subscriber of this Node}}\right)$$

We use a logarithmic scale to still incorporate the importance of node weight without the large channels dominating. I chose this specific one because it helps boost the smaller channels while keeping the bigger channels the same.

3.3.2 Eigenvector Centrality

Eigenvector centrality extends the concept of degree centrality by accounting for the importance of a node's connections. Unlike simpler centrality measures that only

consider the number of connections, eigenvector centrality assigns higher values to nodes connected to other highly connected nodes. The eigenvector centrality x_v of vertex v is defined as:

$$x_v = \frac{1}{\lambda} \sum_{t \in N(v)} x_t \tag{3.11}$$

Where λ is a constant (the eigenvalue) and $N(v)$ represents the set of neighbors of v. This can be rewritten in matrix form as the eigenvector equation:

$$\mathbf{A}\mathbf{x} = \lambda \mathbf{x} \tag{3.12}$$

Where $\mathbf{A}$ is the adjacency matrix of the network.

In our YouTube sports network, channels with high eigenvector centrality represent influential nodes connected to other important channels. These are likely established channels within their respective sports communities that maintain strong connections to other major channels in the same community. By analyzing eigenvector centrality, we can identify the power centers within each sports cluster and understand which channels exert the most influence on the network structure.

Similarly to Betweenness Centrality, we will also be using the same logarithmic scaling to the centrality scores.

3.3.3 Clustering Coefficient

The clustering coefficient quantifies how tightly clustered a node's neighborhood is by measuring the proportion of connections among a node's neighbors relative to the maximum possible number. For a node v with k_v neighbors, the local clustering coefficient C_v is defined as:

$$C_v = \frac{2 \times |\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{k_v(k_v - 1)} \tag{3.13}$$

Where N_i is the neighborhood of vertex v and E is the set of edges in the network.

The global clustering coefficient for our entire network is calculated as the average of all local clustering coefficients:

$$C = \frac{1}{n} \sum_{i=1}^{n} C_i \tag{3.14}$$

In our YouTube sports community analysis, channels with high clustering coefficients are deeply embedded within cohesive communities where many channels share commenters amongst themselves. Conversely, channels with low clustering coefficients may connect to channels from different communities that don't necessarily connect to each other. By analyzing the distribution of clustering coefficients across different sports categories, we can quantify the cohesiveness of different sports communities and identify which sports foster more tightly-knit fan bases on YouTube.

Chapter 4

Visualizations and Findings

In this chapter, we will go over the visualizations and results of the methodology laid out in Chapter 3. We will showcase the two different layouts of the same network we got from ForceAtlas2 and YiFan Hu and go over whether the parameters we used for the Louvain Modularity Algorithm visually found the same clusters or if we have to adjust the parameters. Using our preexisting knowledge as athletes, we will judge whether our visualizations make sense or if there are discrepancies with our knowledge. We hope that between the graph visualizations and additional network metrics from section 3.3, we can find the answers to our research questions.

4.1 Network Visualization Results

We first created the two visualizations colored by the base parameters we stated in section 3.2.1. The nodes are colored by the modularity class and they are sized based on the number of subscribers they have. The edge thickness is related to the number of shared commenters between the two channels, with thicker connections meaning more shared commenters between the channels.

4.1.1 ForceAtlas2 Layout

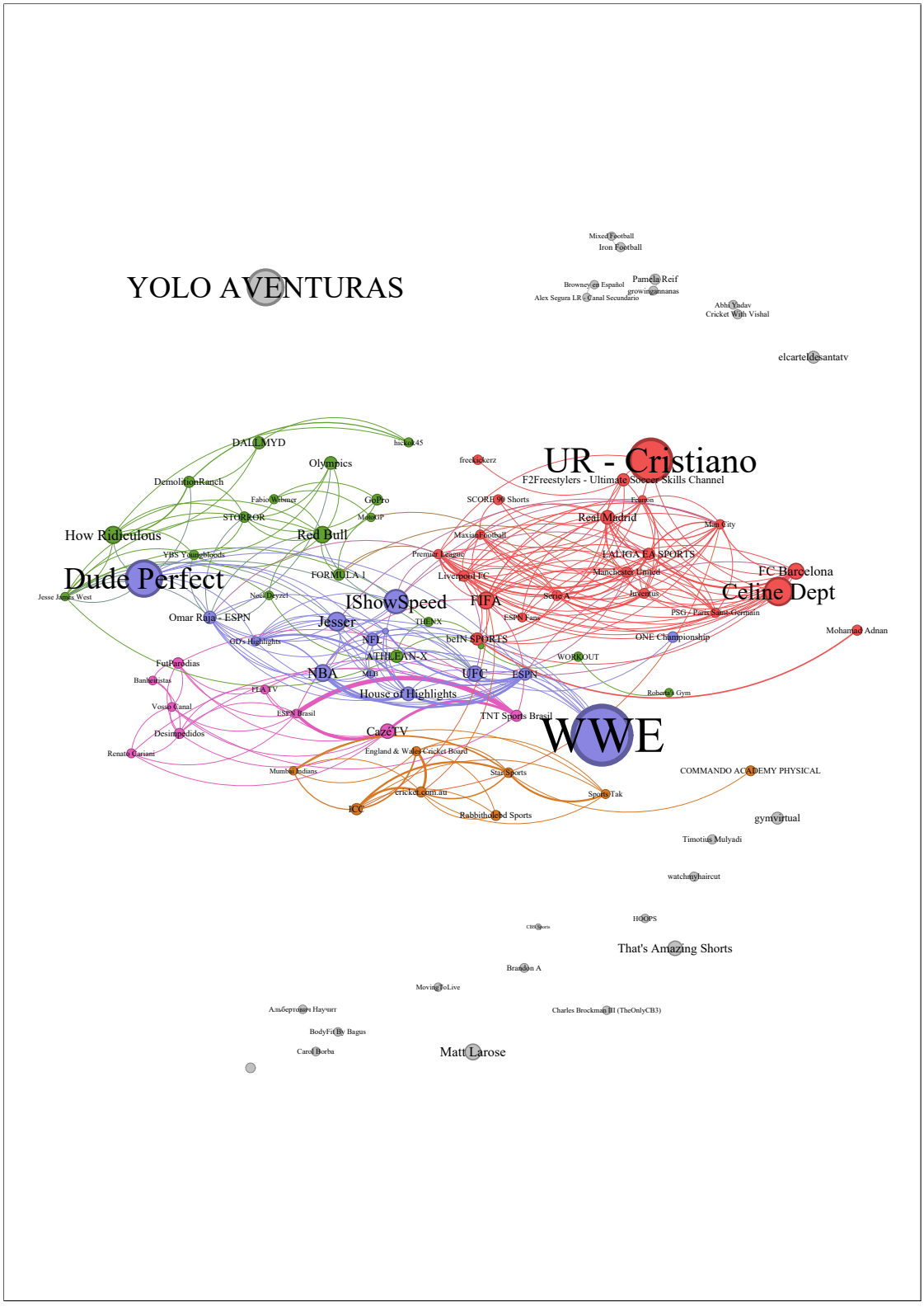

Figure 4.1: ForceAtlas2 (Post Processed) with Louvain Clustering Resolution 1.0

ForceAtlas2 did an extremely good job when looking at the network visually. The disconnected components (or nodes with only links to each other) were spread far apart from the main connected component, so I put the disconnected nodes around the central cluster. The gray nodes were more clustered than the final visualization made it to be, but we decided to put the nodes more spaced out around the edges of the network to emphasize that they didn't have any connections.

There are very clear clusters; however, the post-processing made the clusters more connected than it should be. Below (with just ForceAtlas and NoOverlap), you can see the clusters even more clearly. Notice that without the coloring from the clusters, it would be really hard to discern more than 3, maybe 4 clusters from the pre-processed graph. Also the disconnected nodes were so far away from the main part of the graph I decided not to include it in the preprocessed snippet.

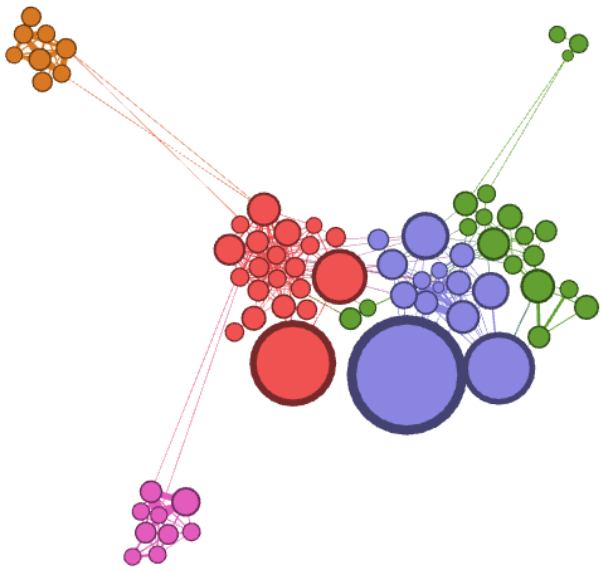

Figure 4.2: ForceAtlas2 (Preprocessed) with no labels and Louvain Modularity Resolution 1.0

4.1.2 Yifan Hu's Proportional Layout

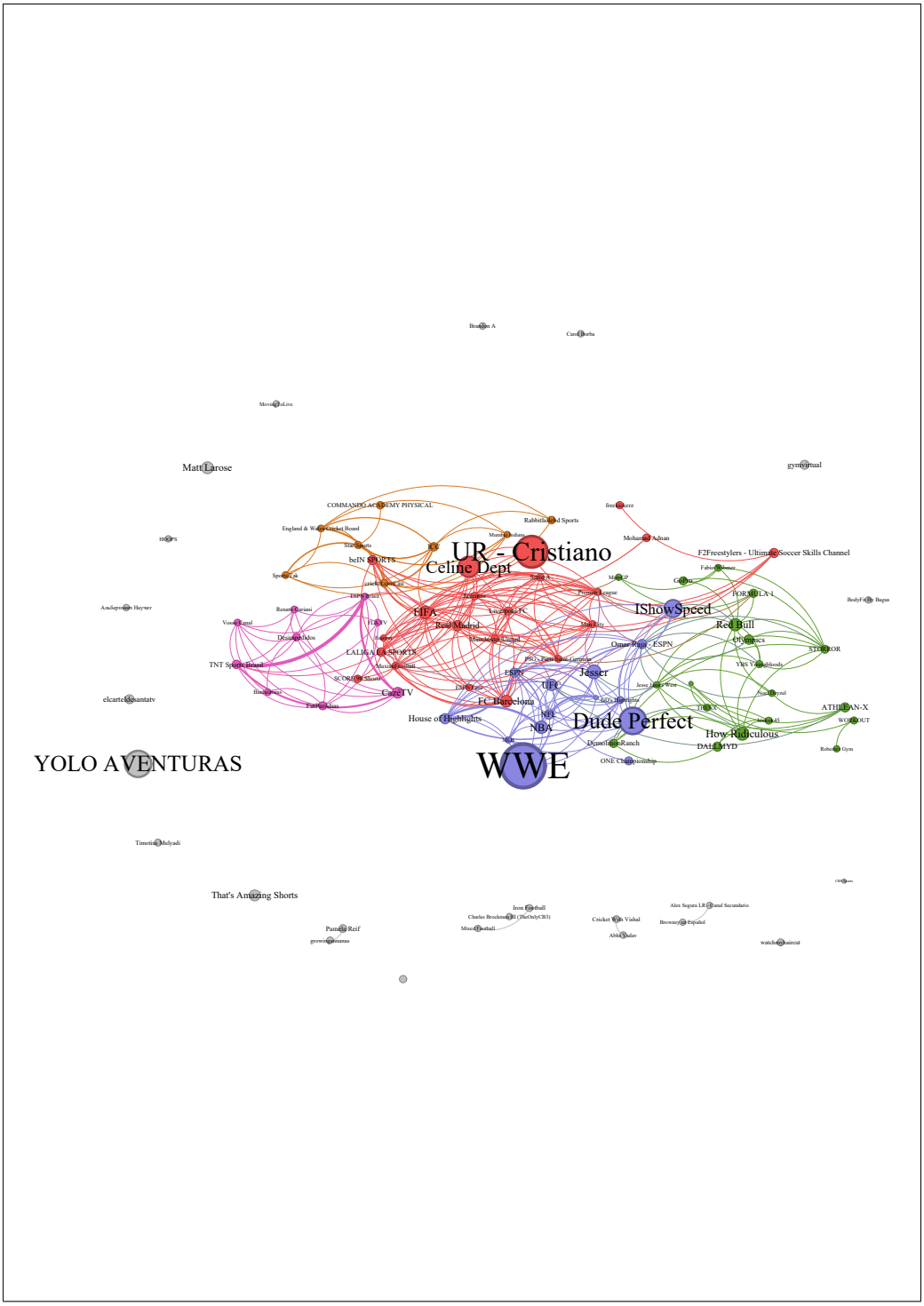

Figure 4.3: YiFan Hu's Proportional (Post Processed) with Louvain Clustering Resolution 1.0

Similar to ForceAtlas2, YiFan Hu's Proportional did a good job visualizing the network. One thing to note is that I did not have to manually move the gray, disconnected nodes, as the algorithm placed them in exactly the spot you see them in the PDF, which I think is helpful in terms of readability.

Similar to ForceAtlas2, the post-processing adjustments such as Contraction and Label Adjust made the graph more connected than it seems.

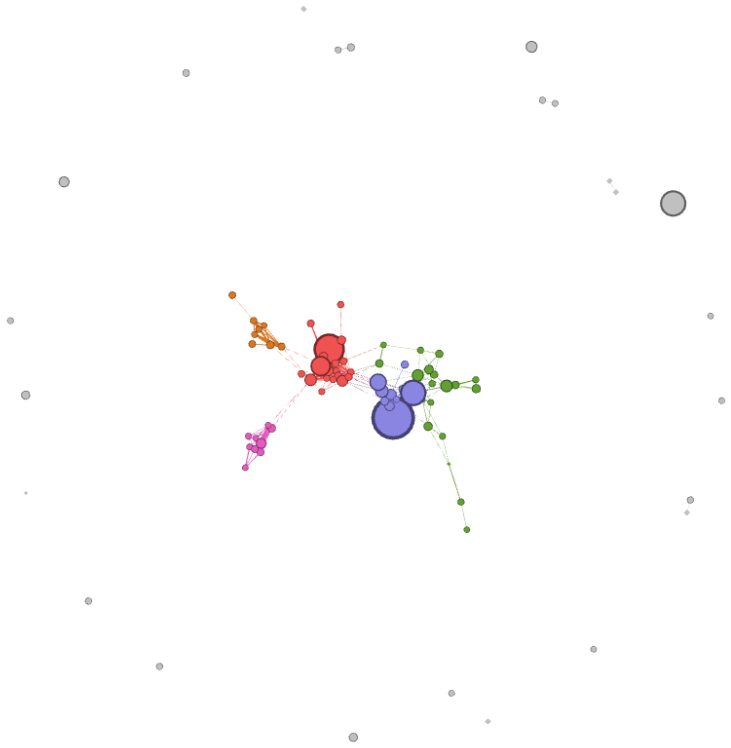

Figure 4.4: YiFan Hu's Proportional (Pre-processed) layout with no labels and Louvain Modularity Resolution 1.0

4.1.3 Comparative Assessment of Layout Algorithms

Both layout algorithms, ForceAtlas2 and Yifan Hu's Proportional, provided valuable visualizations of the YouTube sports channel network based on shared commenters, revealing distinct structural features (Figures 4.1-4.4). However, they exhibited differ-

ent strengths and visual emphases stemming from their underlying mechanics, leading to complementary perspectives on the network's structure.

As initially observed, ForceAtlas2 tended towards generating more compact, densely packed clusters, while Yifan Hu's Proportional algorithm often yielded a more evenly distributed layout across the visualization space. This difference can be directly attributed to their core principles. ForceAtlas2, particularly with the LinLog mode enabled (as discussed in Section 3.1.2), modifies the repulsive force (Eq. 3.4) to decrease more slowly with distance. This function strongly emphasizes the separation between distinct groups of nodes while enhancing cohesion within those groups, naturally leading to visually well-defined and compact clusters as seen in Figure 4.2 (pre-processing). Conversely, Yifan Hu's algorithm (Section 3.1.3), optimizing a global energy function (Eq. 3.1-3.3) often combined with a multilevel approach, seeks a balance between minimizing edge lengths (attraction) and distributing nodes (repulsion) across the available space. This can result in a layout (Figure 4.4, pre-processing) where clusters, while still present, may appear less densely packed or have less sharply defined boundaries compared to ForceAtlas2.

A crucial assessment criterion is how effectively each layout visually represents the community structures identified quantitatively by the Louvain modularity algorithm (results to be detailed in Section 4.2). ForceAtlas2's inherent tendency towards cluster separation often produces a layout where the Louvain-detected communities (represented by node color) appear highly spatially coherent and visually distinct from one another. This provides strong visual reinforcement of the modularity calculations. Yifan Hu's layout, while generally grouping nodes of the same calculated community, sometimes presents these clusters with softer boundaries or allows for slightly more visual inter-mixing at the peripheries, reflecting its different optimization goals. For

example, looking at the blue and green clusters in Figure 4.4 (YiFan Hu's layout), the boundaries between blue and green are not very clear.

Furthermore, the layouts differ in how they highlight other key network features relevant to our research questions:

- **Bridge Channels:** The more pronounced spatial separation between clusters in ForceAtlas2 can make potential "bridge" channels – those connecting disparate communities (to be formally identified via Betweenness Centrality in Section 4.3.1) – more visually apparent, as they might occupy the intermediate space between distinct clusters. Identifying such nodes in the Yifan Hu layout might require more explicit reliance on the centrality metrics, as the spatial gaps may be less pronounced.
- **Edge Weights:** Both algorithms factor in edge weights (shared commenters) to determine attractive forces, positioning strongly connected nodes closer together (visible as thicker edges connecting nearby nodes). Qualitatively, both layouts seemed to represent these strong ties effectively through proximity, though the relative scaling of distances based on weight might differ subtly between the algorithms.
- **Readability and Aesthetics:** Yifan Hu's algorithm sometimes produced an initial layout with slightly better spacing and potentially less initial label overlap before post-processing adjustments. It also handled the placement of disconnected components automatically within the main view (Figure 4.4), whereas ForceAtlas2 often placed them extremely far apart, requiring manual repositioning for the final visualization (Figure 4.2).

Finally, it's essential to acknowledge the impact of post-processing adjustments (Noverlap, Contraction, Label Adjust - Section 3.1.4). While crucial for readability in the

final figures (Figures 4.1, 4.3), these steps, especially Contraction, tended to pull nodes closer to their cluster centers, visually enhancing the cohesion detected by Louvain but also somewhat diminishing the raw layout differences between the two algorithms (compare pre-processing Figures 4.2 and 4.4 with post-processing Figures 4.1 and 4.3).

In conclusion, ForceAtlas2 excels at visually emphasizing the network's modularity, providing strong visual confirmation of community structures aligned with the Louvain analysis. Yifan Hu offers a more balanced global distribution that can aid in overall readability but may require closer integration with calculated metrics to discern subtle structural roles like bridging. For the specific goal of understanding distinct community ecosystems and their separation within the YouTube sports landscape, ForceAtlas2's output, particularly before extensive post-processing, offers compelling visual evidence supporting the concept of distinct community clusters based on shared commenter activity. Both, however, serve as valuable tools for exploring the complex relationships within the network.

4.2 Community Detection Results

To identify distinct community structures within the network of YouTube sports channels, we applied the Louvain modularity algorithm, as detailed in Section 3.2.1. This algorithm groups nodes (channels) based on the density of connections (shared commenters) compared to what would be expected in a random network. The primary analysis was conducted using a resolution parameter of 1.0, utilizing edge weights, and enabling randomization for robustness.

4.2.1 Communities at Resolution 1.0

Running the Louvain algorithm with a resolution of 1.0 on our network graph yielded a total of 25 distinct communities. A significant portion of these represent isolated channels or very small disconnected components that did not meet the 100 shared commenter threshold for forming edges with the main network structure.

Isolated Components

Analysis of the column numbers after sorting the modularity class values reveals the disconnected components of our network. These nodes/small groups (e.g., YOLO AVENTURAS, Matt Larose, pairs like Alex Segura LR and Browney en Español) represent channels whose commenter base, within our sampling limit, did not overlap sufficiently with any other channel in the dataset to form a connection. While they are part of the broader sports landscape on YouTube, they do not appear integrated into the main commenter-sharing network defined by our threshold.

Label	Modularity ...	Interval
MovingToLive	24	
Charles Brockman III (TheOnlyCB3)	23	
HOOPS	22	
BodyFit By Bagus	21	
Альбертович Научит	20	
Carol Borba	19	
Brandon A	18	
Timotius Mulyadi	17	
watchmyhaircut	16	
□□□□ □□□	15	
elcarteldesantatv	14	
gymvirtual	13	
That's Amazing Shorts	12	
Matt Larose	11	
YOLO AVENTURAS	10	
CBS Sports	9	

Figure 4.5: Snippet of the Disconnected Nodes in our network

Communities within the Main Connected Component

Focusing on the modularity numbers that have the most nodes, we can identify distinct communities at our default resolution of 1.0. These clusters represent groups of channels with significantly higher internal commenter overlap than external overlap with other clusters. I will reference these clusters by the coloring scheme I used in all the layout algorithms in section 4.1 to refer to these communities. The primary communities are:

- **Community Red: European Football Leagues and Global Soccer**
  - *Description:* This large community seems centered around European soccer teams/leagues, global football governing bodies, and high-profile international channels. It likely represents the core global soccer fandom accessible through YouTube.
  - *Key Channels:* The two biggest nodes include Cristiano and Celine Dept. Cristiano being regarded as one of the greatest European soccer players in the world and as one of the world's most famous men, and Celine Dept is a former professional football player turned influencer with over 46.6 million subscribers on YouTube at the time of this project. Another important node to mention is FIFA, as it has a central location in both layout algorithms. This is due to its high betweenness centrality score, which will be elaborated on in section 4.3.
  - *Visual Representation:* The cluster is central in YiFan Hu's layout algorithm (Figure 4.3) and top right in ForceAtlas2 (Figure 4.1)
- **Community Blue: US Sports and Highlight Culture**
  - *Description:* This community features major US sports leagues (NBA, NFL, MLB), general sports broadcasters (ESPN), and popular highlight/personality-

driven channels. It showcases significant overlap between official league channels and broader sports entertainment content, giving insight into the "culture" of American Sports and its content.

  - *Key Channels:* The two visually biggest channels in this cluster are Dude Perfect and WWE. Dude Perfect is an influencer group that got famous doing trick shots and collaborations with professional athletes. WWE is the largest pro wrestling promotion in the world. However, another important node is Jesser, with its central location in both layout algorithms, it is bound to have a high centrality score (which we will explore in the next section). Note how there is a mix of official leagues and personality channels.
  - *Visual Representation:* The cluster is central in both layout algorithms with YiFan Hu's layout putting it more center-right.

- **Community Green: Fitness, Extreme Sports and Lifestyle**
  - *Description:* This cluster brings together channels focused on fitness training, extreme sports, and related lifestyle content. It highlights an audience interested in physical performance and adventure content.
  - *Key Channels:* Some key channels in this include How Ridiculous, Red Bulls, Athlean-X.
  - *Visual Representation:* For YiFan Hu's layout this cluster is located on the right and for ForceAtlas2's layout, this is located on the top left. Notice how this cluster has close proximity and is always next to the blue cluster in both layout algorithms.

- **Community Magenta: Brazilian Sports/Soccer Focus**
  - *Description:* This community appears dominated by channels popular in

Brazil, heavily featuring soccer content, possibly including lifestyle/fitness crossover.

  - *Key Channels:* Prominent members include TNT Sports Brasil, CazéTV, and ESPN Brasil.
  - *Visual Representation:* The cluster is on the bottom left for both layout algorithms.

- **Community Brown: Cricket Focus**
  - *Description:* This community clearly clusters around the sport of Cricket and its international and regional bodies.
  - *Key Channels:* Prominent channels are ICC, cricket.com.au, Star Sports, and England and Wales Cricket Board.
  - *Visual Representation:* This cluster is at the bottom center for the ForceAtlas2 layout and top left for the YiFan Hu's layout algorithm.

Discussion of Resolution 1.0 Findings:

At resolution 1.0, the Louvain algorithm successfully identified several thematically coherent communities within the main network component. The groupings largely align with intuitive understandings of sports fandom (e.g., distinct clusters for European soccer, US sports, Cricket, Fitness). The visualization provided by ForceAtlas2 (Figure 4.1/4.2), in particular, emphasizes the separation of clusters magenta and brown; however did a great job making clusters red, blue, and green identifiable while indicating they have a lot of crossover fans/community. YiFan Hu's layout algorithm does a good job with spacing the nodes and clusters in general (as mentioned in section 4.1.3).

4.2.2 Exploring Different Resolutions

To assess the robustness of the community structure and explore different levels of granularity, we also ran the Louvain algorithm with varying resolution parameters.

Higher Resolution (2.0)

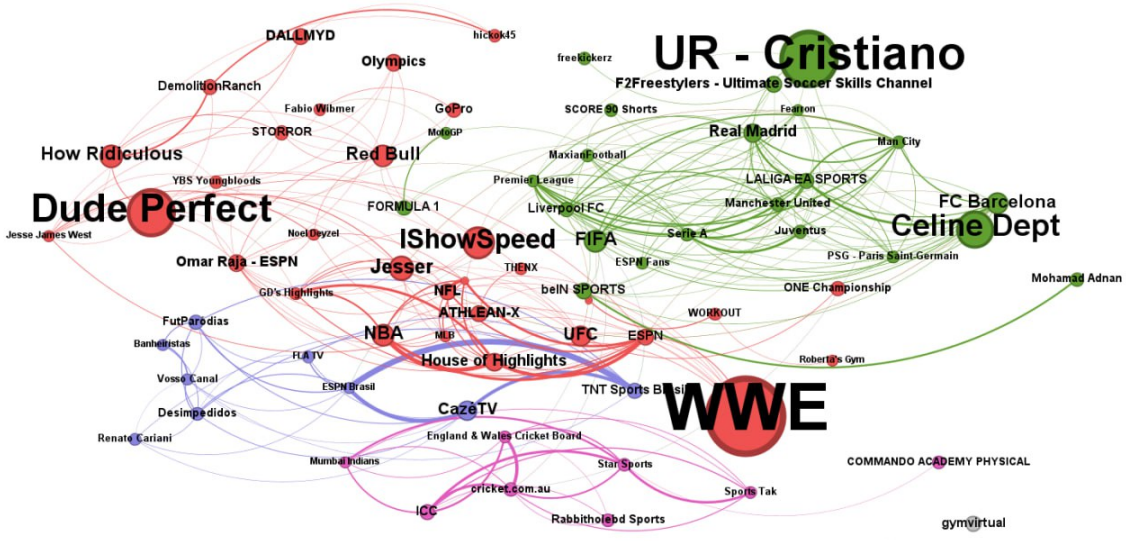

Figure 4.6: Snippet of ForceAtlas2 layout with Louvain Clustering Resolution 2.0

Running Louvain with resolution 2.0 resulted in combining the original clusters, green and blue, to be one giant cluster. I believe this is because the extreme sports and lifestyle channels have lots of similarities (i.e Dude Perfect vs. How Ridiculous and WWE and Red Bull), so their communities would be the ones most likely to merge.

Lower Resolution (0.5)

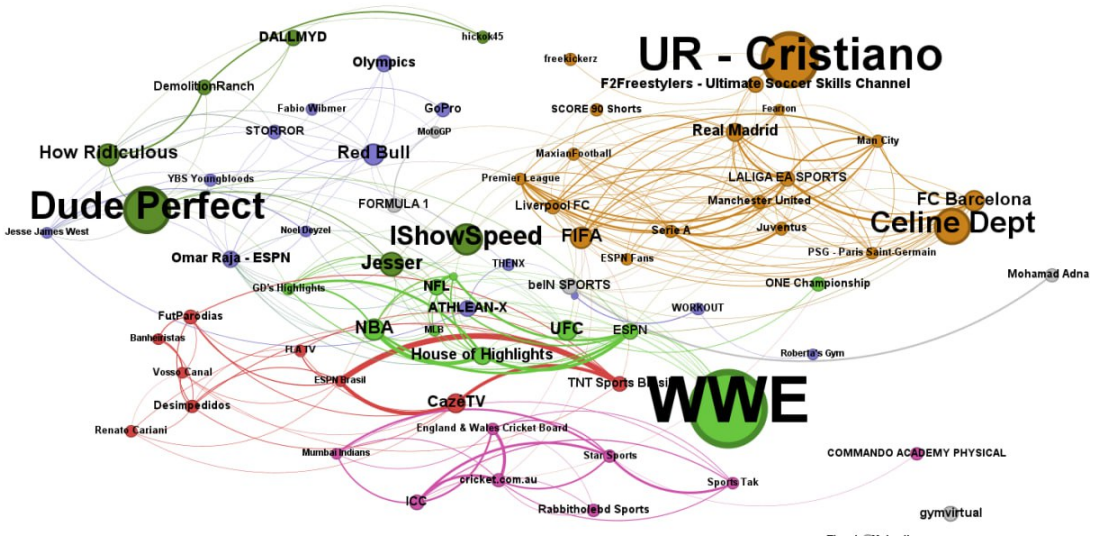

Figure 4.7: Snippet of ForceAtlas2 layout with Louvain Clustering Resolution 0.5

Using resolution 0.5 yielded 3 more communities than the original 25; however, the more interesting detail is that this resolution created 3 clusters from the original green and blue clusters. The three new clusters being:

- Dark Green: Separated the original sports personalities into their own cluster (Dude Perfect, Jesser, IShowSpeed).
- Light Green: This cluster is more strict being official sports channels (MLB, NFL, NBA, UFC) and their corresponding highlights channels (House of Highlights, ESPN, GD's Highlights).
- Blue: Similar to our original green cluster, this cluster represents extreme sports and fitness channels

Discussion of Resolution Impact:

Comparing the community structures across different resolutions confirms that our original clusters, red, magenta, and brown, are relatively stable. By changing the resolution, you can tell that there are a lot of nuances between American communities (original clusters green and blue), whereas fans from other countries are more dedicated to their specified niches (European soccer, cricket, and Brazilian sports).

4.2.3 Section Summary:

The community detection analysis revealed a core network structure composed of 5 distinct communities within the main connected component, surrounded by numerous isolated channels/groups, when analyzed at a baseline resolution of 1.0. These core communities largely correspond to major global sports (Soccer, Cricket), regional sports markets (the US, UK, Brazil), and thematic interests (Fitness/Lifestyle). Exploring different resolutions confirmed the general robustness of these core clusters while also highlighting potential sub-structures and higher-level associations. The identified communities provide a data-driven map of how major sports fandoms interact and segment based on shared audience engagement patterns on YouTube.

4.3 Centrality and Clustering Coefficient Results

Beyond the visual layout and community detection, analyzing network centrality and clustering coefficients provides quantitative insights into the structural roles and cohesiveness of channels within the YouTube sports ecosystem. We focus on Scaled Betweenness Centrality to identify bridges, Scaled Eigenvector Centrality to identify influential hubs, and the Clustering Coefficient to measure local density, using the scaled metrics derived as described in Section 3.3 to account for channel size differences.

4.3.1 Identifying Bridge Channels (Scaled Betweenness Centrality)

Scaled Betweenness Centrality quantifies how often a channel lies on the shortest paths connecting other channels, highlighting key bridge channels. Channels scoring high on this metric bridge different communities or topics.

The analysis revealed several notable bridges, including:

- **FIFA** (Scaled Betweenness $\approx$ 1257.0) and **Premier League** ($\approx$ 1139.7) show extremely high scores, reflecting their roles connecting numerous global and regional football channels.
- Major Broadcasters/Aggregators like **Omar Raja - ESPN** ($\approx$ 613.5), **ESPN Brasil** ($\approx$ 478.0), **TNT Sports Brasil** ($\approx$ 426.0), and **UFC** ($\approx$ 627.4) also act as crucial bridges, likely due to their coverage spanning multiple sports, teams, or types of content (highlights, news).
- Interestingly, **cricket.com.au** ($\approx$ 585.7) emerges as a strong bridge, likely connecting various international and regional cricket communities.
- Fitness channel **ATHLEAN-X** ($\approx$ 345.5) and personality-driven channel **Jesser** ($\approx$ 309.4) also show notable betweenness scores, suggesting they connect fitness or highlight/entertainment spheres with core sports communities.

These channels often appear visually positioned between the major clusters (or within their own cluster) identified in Section 4.2, confirming their role in facilitating connections across the network.

4.3.2 Identifying Influential Hubs (Scaled Eigenvector Centrality)

Scaled Eigenvector Centrality measures influence, giving higher scores to channels connected to other influential channels. High scores indicate hubs central to well-connected communities.

The most influential channels are dominated by major European football entities:

- **Premier League** (Scaled Eigen Centrality ≈ 2.12) has the highest score, followed closely by major clubs like **Man City** (≈ 1.97), **Juventus** (≈ 1.78), **LALIGA EA SPORTS** (≈ 1.74), **FC Barcelona** (≈ 1.66), **Real Madrid** (≈ 1.65), **Manchester United** (≈ 1.65), and **Liverpool FC** (≈ 1.63).
- **FIFA** (≈ 1.54) and **Serie A** (≈ 1.57) also rank highly.

These channels are predominantly members of the 'Red' European Football/Global Soccer community identified in Section 4.2. Their high scores confirm their status as central, highly interconnected players within this dense part of the network, rather than necessarily bridges between disparate parts (though some, like the Premier League and FIFA, score highly on both metrics).

Some other influential nodes within the other clusters include:

- **Omar Raja - ESPN** (Scaled Eigen Centrality ≈ 0.94) - Belongs to the 'Blue' community (Class 6). Its high score suggests it's a very influential node connecting to other important channels within the US sports and highlights sphere.
- **GD's Highlights** (≈ 0.86) - Also in the 'Blue' community (Class 6). Similar to Omar Raja, it acts as an influential aggregator/hub within that cluster.

- **ESPN** (≈ 0.83) - The main ESPN channel, also 'Blue' (Class 6), serves as a major influential node connecting various US sports entities.
- **Jesser** (≈ 0.73) - 'Blue' community (Class 6). This personality-driven channel shows significant influence, connecting well with other major players in the US sports entertainment/influencer space.
- **UFC** (≈ 0.70) - 'Blue' community (Class 6). As a major sports organization, it ranks as an influential hub within the broader US sports/combat sports cluster.

So, while European football channels form the most dominant influential core based on this metric, there are clearly influential hubs within the US-centric sports and entertainment community ('Blue') as well, represented by major broadcasters, highlight channels, and key personalities.

4.3.3 Local Cohesion (Clustering Coefficient)

The Clustering Coefficient (CC) measures how interconnected a channel's immediate neighbors are. A score near 1 indicates a tight-knit local environment, while a score near 0 suggests connections to diverse, unrelated channels.

The network exhibits a wide range of local cohesion:

- Numerous channels have a CC of **1.0**, including **FLA TV**, **Renato Cariani**, **SCORE 90 Shorts**, **Mumbai Indians**, **Sports Tak**, **Banheiristas**, **ESPN Fans**, **GoPro**, **Olympics**, and **hickok45**. This indicates they exist within very specific niches or closed triads where their direct connections strongly overlap. Many of these belong to the Brazilian (Magenta) or Cricket (Brown) communities, or represent distinct interests.
- Conversely, many channels have a CC of **0**, including several fitness channels (**Roberta's Gym**, **COMMANDO ACADEMY PHYSICAL**), iso-

lated language-specific channels (**Alex Segura LR - Canal Secundario**, **Browney en Español**), some topic-specific channels (**MotoGP**, **freekickerz**, **ONE Championship**), and nodes identified as isolates in Section 4.2. This suggests they connect to channels that don't share commenters amongst themselves, potentially acting as local bridges or sitting at the periphery.

- Channels with intermediate scores (e.g., **UFC** ≈ 0.35, **Omar Raja - ESPN** ≈ 0.37, **Red Bull** ≈ 0.33) often include the bridge channels identified earlier, consistent with connecting less related neighbors.

This variation in local density helps characterize the different community structures, with some (like European Football, Brazilian Soccer, Cricket) showing higher internal clustering than others (like general US Sports/Highlights or Fitness/Lifestyle).

Synthesis

The centrality and clustering analysis quantitatively confirms the roles suggested by the network visualization. Key organizations (FIFA, Premier League) and broadcasters act as vital bridges (high Betweenness), while major European football clubs form the influential core of the largest community (high Eigenvector). The clustering coefficient further differentiates communities, revealing tightly-knit niches (high CC) alongside channels connecting more diverse audiences (low CC). These metrics provide a deeper understanding of the network's structure and the varied roles channels play within the YouTube sports landscape.

Chapter 5

Implications and Discussion

Our results demonstrate that YouTube sports communities are not self-contained, but instead exist within an interconnected web of shared audiences. The use of the Louvain modularity algorithm allowed us to identify distinct community clusters based on behavioral overlap rather than pre-assumed categories. In doing so, we discovered both expected clusters, like those between soccer fandoms (*evident in the large 'Red' European Football/Global Soccer community*), and surprising cross-sport bridges, like fitness creators (*such as ATHLEAN-X, identified with significant Betweenness Centrality*) connecting fans of MMA, bodybuilding, and general sports content. This result highlights a key strength of our approach. By using real user behavior (commenter overlap) as the basis for analysis, we captured emergent relationships that traditional survey-based or demographic approaches often miss.

The insights uncovered through our network analysis have several practical implications for content creators, digital marketers, and sports organizations. For content creators, understanding their position in the broader YouTube sports network allows for more strategic collaborations and audience growth strategies. Creators situated within tightly clustered communities, *such as those deeply embedded within the 'Red' (European Football) or 'Brown' (Cricket) clusters,* may benefit from reinforcing their

niche appeal. Conversely, those near the boundaries of multiple clusters, or "bridges," *like FIFA or Premier League which scored highly on both Betweenness and Eigenvector Centrality,* have the unique opportunity to expand across fanbases by *strategically and authentically* producing crossover content. Our analysis highlights these bridge channels as key influencers capable of driving engagement across different sports audiences. Creators in these positions can intentionally diversify their content to tap into new viewership without alienating their core base.

From the perspective of sports marketers, the behavioral data underlying our network offers a more authentic representation of fan engagement than traditional demographic or self-reported survey methods. By identifying where audience overlaps naturally occur, marketers can more effectively tailor campaigns and sponsorships to reach audiences with shared interests. For instance, *the observed connections between major league channels in the 'Blue' (US Sports/Highlight) cluster and lifestyle/fitness creators in the 'Green' cluster suggest that* major league marketers have the unique ability to connect with fans through someone who already relates more to them than the average big leaguer. Instead of relying on isolated audience segments, marketers can use this behavioral overlap to design campaigns that speak to broader communities and leverage influencers *like Jesser or Omar Raja - ESPN, identified as bridges or influential hubs,* who already span multiple fanbases.

For sports organizations, particularly leagues and teams with official YouTube presences, the findings suggest that digital engagement is no longer limited to diehard fans of a single sport. Viewers are increasingly fluid in their consumption, often interacting with a mix of entertainment, analysis, fitness, and cross-sport content. *This fluidity reflects broader trends in digital media where users curate personalized content streams across diverse interests.* Organizations that recognize and adapt to this trend can grow their digital footprint by partnering with bridge creators (*perhaps channels like ATHLEAN-X or Red Bull found in the 'Green' cluster*), producing mul-

timedia content that appeals to adjacent communities, or even reshaping their brand to engage audiences across traditional boundaries.

Beyond the technical outcomes, this project underscores the theoretical and practical significance of behavioral network analysis in the digital age. From a theoretical standpoint, our findings challenge views of sports fandom as static and segmented. In contrast, we show that fan identities are increasingly fluid, with users engaging across sports, geographies, and content types. Practically, this insight can guide content creators, sports marketers, and media platforms in better targeting and engaging hybrid audiences. *Ultimately, understanding the network structure, knowing which channels are central hubs like Man City, which are vital bridges like FIFA, and which communities like Cricket form tight-knit groups, provides a powerful lens for navigating and succeeding within the modern digital sports landscape.* Understanding which channels are interconnected can inform strategic collaborations, content diversification, and audience growth initiatives.

Chapter 6

Conclusion

Our project set out to uncover and visualize the hidden structures that define YouTube's sports communities by analyzing shared commenter behavior across 95 of the platform's most prominent sports channels. We pulled commenter data from the platform and constructed and visualized our networks according to two force-directed layout algorithms, namely ForceAtlas2 and YiFan Hu's Proportional algorithm. Our findings revealed both the macro-level architecture of sports fandom online, *identifying distinct clusters often based on sport type or region,* and the micro-level connections, *including key bridge channels and influential hubs,* that bind it together.

The comparison between ForceAtlas2 and YiFan Hu's layouts was also informative. ForceAtlas2 offered sharper community delineation due to its clustering emphasis, while YiFan Hu's algorithm produced more evenly distributed layouts that improved label readability and visual balance. The convergence of insights across both visualizations strengthens the robustness of our findings and supports the validity of our network structure.

That said, our project was not without limitations. API rate limits, language inconsistencies, and the exclusion of private or deleted commenter accounts may have reduced our dataset's completeness. Additionally, the reliance on commenter IDs does

not capture passive engagement (e.g., viewers who watch but do not comment), which likely constitutes a large share of the audience. We also acknowledge the presence of potential bot activity and inflated engagement metrics on some channels, which we attempted to mitigate through a thresholding strategy but could not fully eliminate.

Future research could build on this foundation. *To address the static snapshot nature of our current analysis,* incorporating seasonality tracking could reveal how these communities evolve over time. Certain communities may engage with others more prominently in offseason phases. For example, baseball fans may interact with basketball and football pages more while baseball is in the offseason (November-April) and other sports are in full swing, entering playoffs. Including sentiment analysis can also lead to better understanding of the tone and content in cross-community interactions; perhaps some sports gel better with others while others may exhibit some distaste. *Furthermore, to move beyond comment-only data,* incorporating multimodal engagement metrics (likes, shares, watch time) and cross-platform comparisons (e.g., TikTok, Instagram) would also add depth to our understanding of how sports communities form and evolve online.

In conclusion, this project not only demonstrates the value of using network science to decode the digital sports environment, but also contributes to a growing body of research showing that online fan behavior is complex, dynamic, and deeply interconnected. YouTube is not just a platform for sports consumption; it is a living map of modern fandom, shaped by content and community.

Bibliography

[1] Y. Hu. Efficient and high quality force-directed graph drawing. In *Proceedings of the 13th International Symposium on Graph Drawing*, pages 167–178. Springer-Verlag, 2005.

[2] M. Jacomy, S. Heymann, T. Venturini, and M. Bastian. Forceatlas2, a continuous graph layout algorithm for handy network visualization. 2012. Draft.

[3] Social Blade. Most subscribed sports youtube channels, 2025. https://socialblade.com/youtube/top/category/sports/mostsubscribed.

[4] Social Blade. Top 100 sports youtube channels sorted by subscribers, 2025. https://socialblade.com/youtube/lists/top/100/subscribers/sports/global.