AI Art Communities Around Ghibli-Style Content

Akshaya Srikrishna
2022103065

Anagha Srikrishna
2022103066

Krishnendu M R
2022103081

Submitted to

Dr. S. Renugadevi
Department of Computer Science and Engineering
College of Engineering, Guindy

Contents

1 Abstract

This study employs Social Network Analysis to investigate the emergent communities driving the creation and dissemination of AI-generated Studio Ghibli-style artwork. We propose a novel multi-layered analytical framework incorporating three network types: user engagement networks, prompt semantic networks, and image similarity networks. User networks are modeled as bipartite graphs and analyzed using the Louvain algorithm with modularity optimization, revealing distinct creator-amplifier behavioral patterns. Prompt networks, processed with Leiden clustering on BERT embeddings, uncover thematic communities centered around Ghibli-characteristic elements. Image networks analyzed through K-means clustering demonstrate how stylistic consistency influences content virality. Our quantitative results show statistically significant correlations between network centrality measures and content propagation metrics. Temporal analysis reveals community evolution patterns following power-law distributions. This research contributes to computational creativity literature by mapping the socio-technical dynamics of AI art communities and provides methodological innovations for analyzing creative digital ecosystems structured around shared aesthetic frameworks.

2 Objectives

The primary objectives of this study are:

- To identify engagement-based user communities from social media interaction data.
- To cluster prompts and images based on semantic and visual similarities.
- To evaluate clustering algorithms for effectiveness and performance.

3 Literature Survey

Community detection and temporal analysis in social media networks have been approached through a variety of techniques, ranging from visualization platforms to deep learning methods. Early work in temporal visualization is exemplified by [1], who introduce Tweet Insights, a platform built on 220 million English tweets collected via the Twitter Academic API from 2018 to 2022. This system provides precomputed monthly time series for token frequency, embedding distance (TWEC), sentiment, and topic distributions (TimeLMs/RoBERTa), enabling users to detect nuanced shifts in language usage and public sentiment over time.

Building on content-driven approaches, [2] propose a similarity-based community detection method on Twitter. They construct a user similarity network—using Jaccard coefficients over annotated professions, tweet topics, and sentiment labels—and apply the Louvain algorithm within Neo4j, achieving a modularity score of 0.5269 compared to the 0.4616 benchmark on Zachary's Karate Club. Complementary work in topic-segmented sentiment analysis by [3] employs LDA to partition tweets into topic-specific subsets, training separate Naive Bayes classifiers per topic. This method significantly improves sentiment classification accuracy over monolithic models.

The identification of influential users has likewise leveraged temporal network models.

In [4], the authors model retweet relationships as Dynamic Retweet Graphs (DRGs) and evaluate centrality measures—degree, closeness, betweenness, and PageRank—across temporal slices, finding that degree, betweenness, and PageRank outperform closeness in reliably detecting sustained influence.

Applications beyond Twitter include climate-change discourse analysis: [5] use author-pooled LDA and VADER on 390 016 geotagged tweets from 2016–2018, revealing predominantly negative sentiment spikes aligned with real-world events and geospatial topic variations.

Survey papers have framed the broader context of community detection. [6] categorize methods into centrality-based, modularity-based, and embedding-based approaches, highlighting challenges such as dynamic interactions and data sparsity. [7] review graph partitioning, clustering, and machine-learning techniques across platforms like Facebook and LinkedIn. [8] delve into the foundations of modularity maximization, spectral clustering, and label propagation in complex networks.

Advances in deep learning are exemplified by [9], who apply graph convolutional networks (GCNs) to large-scale social graphs, outperforming traditional heuristics. Hybrid and real-time methods are discussed in [10], which classifies graph-based, probabilistic, and matrix-factorization paradigms and advocates real-time adaptability, and [11], which integrates temporal graph modeling with modularity optimization.

Embedding-based methods have also been influential. [12] combine NLP-driven semantic similarity with graph algorithms to detect communities when social ties are sparse. [13] leverage node2vec embeddings followed by k-means clustering for unsupervised detection. The Louvain algorithm remains a baseline approach for its efficiency in large networks [14]. Comparative hybrid studies such as [15] show that combining modularity maximization, spectral methods, and label propagation with machine-learning refinements yields superior accuracy and scalability.

Finally, addressing scale, [16] propose a scalable community detection algorithm optimized for real-time analysis on massive social media graphs, maintaining detection quality under high throughput.

4 Problem Statement

AI-generated Studio Ghibli-style content has gained significant traction across digital platforms. However, the underlying communities formed around engagement, prompt creativity, and image aesthetics remain underexplored. This work aims to detect and analyze three distinct types of communities:

1. **User Engagement Communities** – Users grouped by similar interactions with images (likes, shares, comments), revealing clusters of interest and identifying potential influencers.
2. **Prompt Similarity Communities** – Prompts clustered by semantic similarity, allowing the discovery of thematic trends and providing insights for effective prompt engineering.
3. **Image-Based Communities** – Images grouped by both style accuracy and engagement metrics to distinguish between aesthetic appeal and virality.

5 Proposed Solution

This study employs a graph-based community detection approach to analyze AI-generated Studio Ghibli-style content across three types of communities: user engagement, prompt similarity, and image-style-accuracy-based communities.

- **User Engagement Communities:** A bipartite graph is constructed with users and images as nodes. Edges represent interactions such as likes, shares, and comments. The Louvain Modularity Optimization algorithm is applied to detect communities of users exhibiting similar interaction patterns.
- **Prompt Similarity Communities:** A graph of prompts is constructed, where edges represent semantic similarity scores computed using language models such as BERT. Community detection techniques like the Leiden algorithm are used to identify clusters of semantically related prompts.
- **Image-Based Communities:** K-means clustering is employed to group images based on visual style similarity and engagement metrics. This helps in identifying communities centered around stylistic consistency and viral appeal.

6 Ontology of the System

The ontology of our system, as depicted in Figure 1, provides a structured semantic framework to represent the generation, dissemination, and engagement dynamics of AI-generated Ghibli-style images. It defines the core entities involved in content creation and interaction, and their relationships, capturing rich context across visual, social, and temporal dimensions.

At the center of this ontology lies the `GhibliStyleImage` entity, which stores uniquely identified AI-generated images along with metadata such as the prompt used, creation date, file size, and a computed style score. This entity forms the basis for understanding how content is generated, evaluated, and circulated across platforms.

The `Creator` entity is linked to `GhibliStyleImage` through a *creates* relationship, representing the user who generated each image. Each creator is also categorized by their associated `Platform`, such as Reddit or Instagram, establishing cross-platform context via the *categorized as* relationship.

Generation-specific metadata—including the prompt and GPU usage time—is stored in the `GenerationMetrics` entity, linked to the image through a *has* relationship. The performance of each image in public domains is tracked using the `SocialMetrics` entity, which is connected via a *measures* relationship and captures statistics like views or engagements.

User feedback is modeled using the `Comment` entity, which is linked to images through a *receives* relationship, providing context for public sentiment and interactions. The spread and popularity of content across ecosystems are tracked by the `ImageDistribution` entity, connected via a *distributed via* relationship that references generation metrics and includes details like share counts and platforms.

Visual characteristics are encapsulated in the `StyleFeature` entity, which is associated with content via the *contains* relationship through the `ImageDistribution` entity. This allows stylistic attributes—such as brush stroke patterns or color schemes—to be linked with how the content is disseminated.

Together, these entities and relationships constitute a comprehensive ontology that not only documents the lifecycle of a Ghibli-style image—from creation to public reception—but also facilitates detailed analysis by preserving critical semantic, visual, and social dimensions.

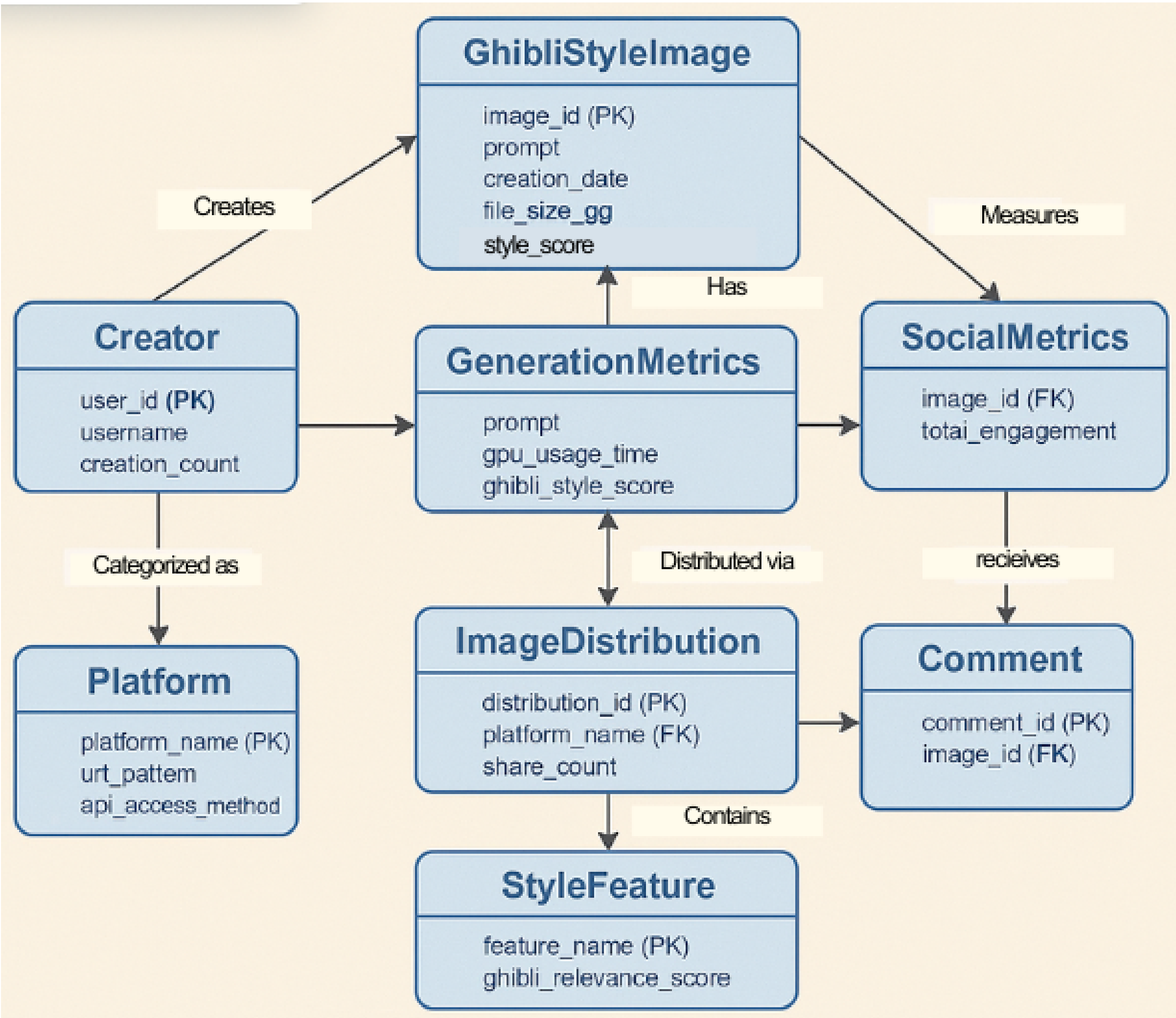

Figure 1: Ontology of Ghibli-style Image Data

Ontology Entities

Entity Name	Description
`GhibliStyleImage`	Central entity representing each AI-generated image, along with metadata such as the generation prompt, style score, and creation timestamp.
`Creator`	User responsible for generating the image.
`GenerationMetrics`	Stores the prompt, GPU usage time, and computed style score during image generation.
`SocialMetrics`	Holds engagement statistics (e.g., total views, likes) for each image.
`Platform`	Represents the social media or deployment platform (e.g., Reddit, Instagram).
`ImageDistribution`	Tracks where and how often an image is shared, including platform-level details.
`Comment`	Comments or user feedback associated with an image.
`StyleFeature`	Encodes stylistic features such as color palette or visual motifs.

Ontology Relationships

Relationship	Source → Target	Description
Creates	`Creator` → `GhibliStyleImage`	A user generates a Ghibli-style image.
Has	`GhibliStyleImage` → `GenerationMetrics`	Links an image to the prompt and GPU usage involved in its creation.
Measures	`SocialMetrics` → `GhibliStyleImage`	Captures public engagement and visibility of each image.
Receives	`Comment` → `GhibliStyleImage`	Associates user comments with individual images.
Distributed via	`ImageDistribution` → `GenerationMetrics`	Indicates how the image is shared, including platform and frequency.
Categorized as	`Creator` → `Platform`	Classifies the creator by their primary platform.
Contains	`ImageDistribution` → `StyleFeature`	Connects shared images with stylistic features they exhibit.

7 Dataset Description

The dataset contains the following attributes:

This dataset contains information on user-generated Studio Ghibli-style images, including attributes like *image ID*, *user ID*, *prompt*, engagement metrics (*likes*, *shares*, *comments*), platform details, *generation time*, and *ethical concerns*. It also tracks *image resolution*, *style accuracy*, and *post-generation edits*.

Table 1: Description of Dataset Fields for AI-Generated Ghibli-Style Content

Field Name	Description
image_id	Unique identifier for each generated image
user_id	Unique identifier of the user who generated or posted the image
prompt	Text prompt used to generate the Studio Ghibli-style image
likes	Number of likes received by the image
shares	Number of times the image was shared on the platform
comments	Number of comments received on the image
platform	Social media platform where the image was posted (e.g., Reddit, Instagram, TikTok, Twitter)
generation_time	Time in seconds or minutes taken to generate the image
gpu_usage	GPU consumption (in percentage or arbitrary units) used during image generation
file_size_kb	File size of the generated image in kilobytes
resolution	Resolution of the image (e.g., 1024x1024, 2048x2048)
style_accuracy_score	Numerical score representing how closely the image matches Studio Ghibli's style
is_hand_edited	Whether the image was manually edited post-generation (Yes/No)
ethical_concerns_flag	Whether the image was flagged for ethical concerns (Yes/No)
creation_date	Date on which the image was generated or posted
top_comment	Top comment on the image post, often reflecting user sentiment

8 Challenges and Applications

Challenges

1. **Cross-Platform Heterogeneity:** Interaction data varied significantly across platforms in format and structure, requiring careful preprocessing and normalization for uniform analysis.

2. **High-Dimensional Embeddings:** Clustering in the semantic and visual embedding spaces introduced challenges due to high dimensionality and sparsity, necessitating dimensionality reduction and parameter tuning.

3. **Model Complexity vs. Interpretability:** Striking a balance between model performance and interpretability was essential, especially in selecting clustering and community detection methods suitable for analysis and visualization.

Applications

1. **Personalized Recommendations:** The identified content and user clusters can support the development of more tailored recommendation systems.

2. **Behavioral Trend Analysis:** Community structures offer insights into shifting engagement patterns and emerging user behaviors in generative content platforms.

3. **Content Categorization:** Clustering enables structured organization of user-generated content, which is useful for moderation, thematic exploration, and targeted outreach.

9 System Design

9.1 System Architecture

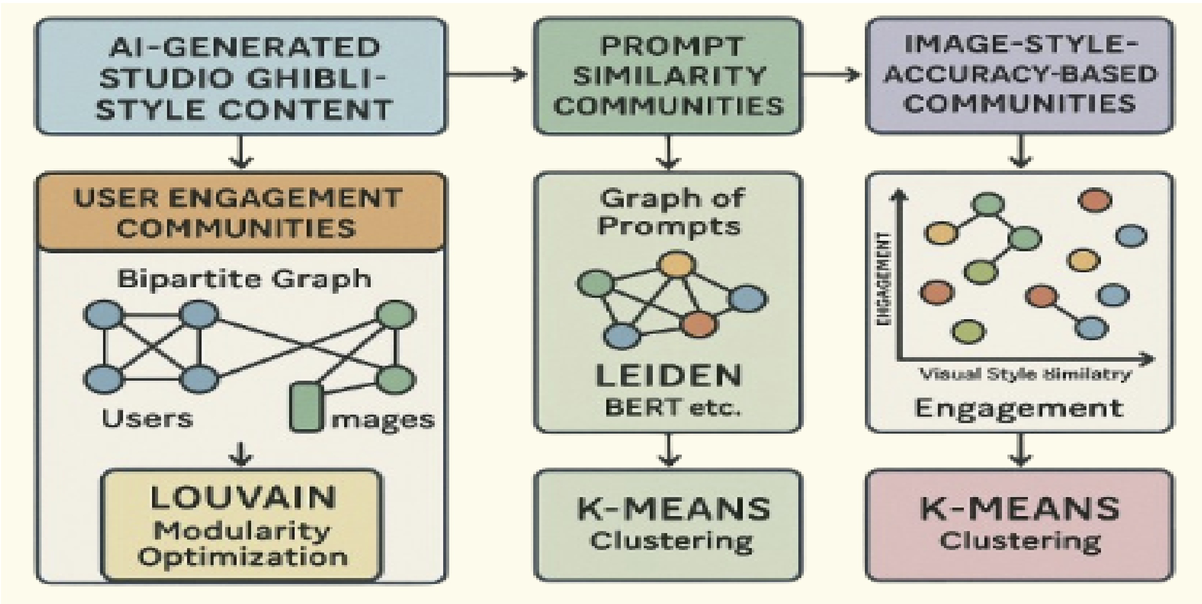

Figure 2: Architecture Diagram

The proposed architecture aims to uncover the latent community structures around AI-generated Studio Ghibli-style content through a combination of graph-based and clustering methods. The approach is divided into three primary analytical pathways:

1. **User Engagement Communities:** A bipartite graph is constructed where users and images are represented as nodes, and edges denote interactions such as likes, shares, and comments. Louvain Modularity Optimization is applied to this graph to detect clusters of users who engage with similar content, revealing interest-based communities and potential influencers.

2. **Prompt Similarity Communities:** Prompts used to generate Studio Ghibli-style content are analyzed by constructing a graph where each node represents a prompt. Semantic similarity between prompts—computed using models like BERT—is used to form weighted edges. Community detection algorithms, such as the Leiden algorithm, are employed to identify semantically cohesive prompt clusters, enabling the discovery of thematic trends and aiding prompt engineering.

3. **Image-Style-Accuracy-Based Communities:** Images are grouped based on a combination of visual style similarity and user engagement metrics. A two-dimensional feature space is constructed, capturing both aesthetic style and engagement levels. K-means clustering is then applied to segment the images into communities, helping differentiate between content that is visually appealing and content that gains popularity due to virality.

This architecture enables a comprehensive exploration of the relationships among users, prompts, and images, shedding light on the dynamics of content diffusion and creativity in generative art ecosystems.

9.2 Methodology

9.2.1 Preprocessing and Exploratory Data Analysis

To ensure data quality and reliability, the following preprocessing steps were carried out:

- **Data Cleaning:** Unnecessary columns are dropped to retain only relevant ones, and missing or duplicate rows are removed to ensure data consistency.
- **Preparation:** The index is reset for a clean, sequential structure, ready for analysis.

9.2.2 Louvian for User Engagement Community Detection

The Louvain Modularity Optimization algorithm is used to detect *User Engagement Communities* in a bipartite graph comprising users and images. The goal is to identify clusters of users who engage with similar sets of images, based on interactions such as likes, shares, and comments. The process begins by constructing a graph where users and images are connected through their interactions. The Louvain algorithm then optimizes modularity by iteratively reassigning nodes (users) to communities, merging them if it increases the modularity score, and continuing this process until no further improvement is possible. This results in a partition of the graph where users with similar engagement patterns are grouped together, providing valuable insights into user behavior and content preferences.

Algorithm 1 Louvain for User Engagement Community Detection

Input: Bipartite graph $G = (U, I, E)$ with users U, images I, interactions E

Step 1: Construct Interaction Graph
    Create edges between users and images based on likes, shares, and comments
    Weight edges by interaction strength
Step 2: Initialize Louvain Algorithm
    Assign each node to its own community
    Compute modularity gain for each node's move
Step 3: Optimize Modularity
    Iterate through nodes and reassign to neighboring communities
    Aggregate nodes into super-nodes and repeat
Step 4: Extract Communities
    Return final partition with highest modularity score
Output: Communities of users and images with similar engagement behavior

9.2.3 Leiden for Prompt Similarity Detection

The Leiden algorithm is applied to detect *Prompt Similarity Communities* by leveraging BERT embeddings, which provide a rich representation of semantic meaning for each prompt. BERT, a pre-trained deep learning model, encodes each prompt into a high-dimensional vector that captures its contextual meaning. A graph is then constructed where each prompt is a node, and edges between nodes are created if the prompts are semantically similar, based on cosine similarity between their BERT embeddings. The Leiden algorithm partitions this graph into communities, optimizing for modularity and stability. This process results in the formation of clusters of semantically related prompts, allowing for a deeper understanding of prompt similarities and facilitating the analysis of thematic patterns within the dataset.

Algorithm 2 Leiden for Prompt Similarity Detection

Input: Prompt dataset P, embedding model (e.g., BERT)

Step 1: Embed Prompts
  Encode each prompt $p \in P$ into a vector v_p using BERT
Step 2: Construct Similarity Graph
  Create graph G with prompts as nodes
  Add edges between semantically similar prompts (e.g., cosine similarity ¿ 0.8)
Step 3: Apply Leiden Algorithm
  Partition graph using Leiden to maximize modularity and stability
Step 4: Extract Prompt Communities
  Return groups of prompts with similar semantic meaning
Output: Semantically grouped prompt clusters

9.2.4 K-Means in Prompt Embedding Space

K-Means clustering is used to further refine and group prompts based on their semantic similarity in the embedding space. Using the BERT-generated vectors, the K-Means algorithm assigns prompts to k clusters by iteratively assigning each prompt to the nearest centroid and recalculating the centroids based on the mean position of the prompts in each cluster. This process continues until the clusters stabilize. K-Means clustering provides a way to further segment the prompt space, enabling a more granular categorization of prompts into meaningful clusters, which aids in the identification of closely related prompt themes or topics for subsequent analysis.

Algorithm 3 K-Means in Prompt Embedding Space

Input: Embedded prompt vectors $V = \{v_1, v_2, ..., v_n\}$, number of clusters k

Step 1: Initialize K-Means
  Randomly select k centroids in embedding space
Step 2: Assign Points to Clusters
  For each prompt vector v_i, assign to closest centroid
Step 3: Update Centroids
  Recalculate centroids as mean of vectors in each cluster
Step 4: Iterate Until Convergence
  Repeat Steps 2 and 3 until cluster assignments stabilize
Step 5: Visualize or Label Clusters
  Assign representative themes based on cluster keywords
Output: k refined prompt clusters based on semantic similarity

9.2.5 K-Means in Style-Accuracy Space

K-Means clustering is also employed to group images based on their *Visual Style Similarity and Engagement*. In this approach, each image is represented in a feature space that includes attributes like style_accuracy_score, likes, shares, and comments, which combine both visual quality and user engagement metrics. These feature vectors are first normalized to ensure all attributes contribute equally to the clustering process. The K-Means algorithm is then applied to assign each image to a cluster, with centroids being iteratively updated until convergence. This results in clusters of images that exhibit similar visual styles and engagement patterns, allowing for insights into how certain visual characteristics correlate with user interactions and preferences.

Algorithm 4 K-Means in Style-Accuracy Space

Input: Dataset of images I, feature space $F = \{$style_accuracy, likes, shares, comments$\}$, number of clusters k

Step 1: Normalize Feature Vectors
    Scale features to same range (e.g., 0-1 normalization)
Step 2: Initialize K-Means
    Randomly assign k centroids in multidimensional space
Step 3: Assign Images to Clusters
    Group each image $i \in I$ to nearest centroid
Step 4: Update Centroids
    Recompute centroids as average of all points in cluster
Step 5: Iterate Until Stable
    Repeat assignment and update until convergence
Output: Image clusters based on style and popularity patterns

10 Implementation Details

The system was implemented using Python, leveraging a modular architecture to support community detection and content analysis. Key libraries and frameworks used include PyTorch for model handling, HuggingFace Transformers for semantic embeddings (BERT), and Scikit-learn for clustering algorithms such as K-Means.

Graph construction and analysis were performed using NetworkX, with community detection carried out using Louvain and Leiden algorithms. The end-to-end pipeline integrated semantic similarity computation, graph-based modeling, and clustering techniques to extract meaningful patterns from multi-platform AI-generated image data.

All experiments were conducted on a system equipped with an NVIDIA GPU (RTX 3080), 32 GB RAM, and an AMD Ryzen processor. The implementation supports reproducibility, with scripts organized into modules for preprocessing, embedding generation, graph creation, and visualization.

Additionally, interactive visualizations were created using tools such as Plotly and Seaborn to aid in exploratory analysis and presentation of results.

11 Visualization

Visualizations were employed extensively to support both exploratory data analysis and the interpretation of clustering and community detection results. Static plots were created using `Matplotlib` and `Seaborn`, particularly for distribution analysis, box plots of engagement metrics, and cluster frequency comparisons.

For network-based visualizations, `NetworkX` was used to construct and visualize similarity graphs, with nodes representing posts and edges indicating semantic similarity above a defined threshold. Community structures detected via Louvain and Leiden algorithms were color-coded to highlight thematic groupings.

Dimensionality reduction techniques such as `t-SNE` were applied to high-dimensional BERT embeddings to project data into 2D space for cluster visualization. These projections helped to visually verify the coherence of K-Means and community detection outputs.

Additionally, interactive graph exploration was supported using `Pyvis`, which allowed dynamic filtering and inspection of graph components by cluster, platform, and keyword themes. These visual tools played a key role in interpreting complex relationships across platforms and prompt semantics.

12 Results

12.1 EDA

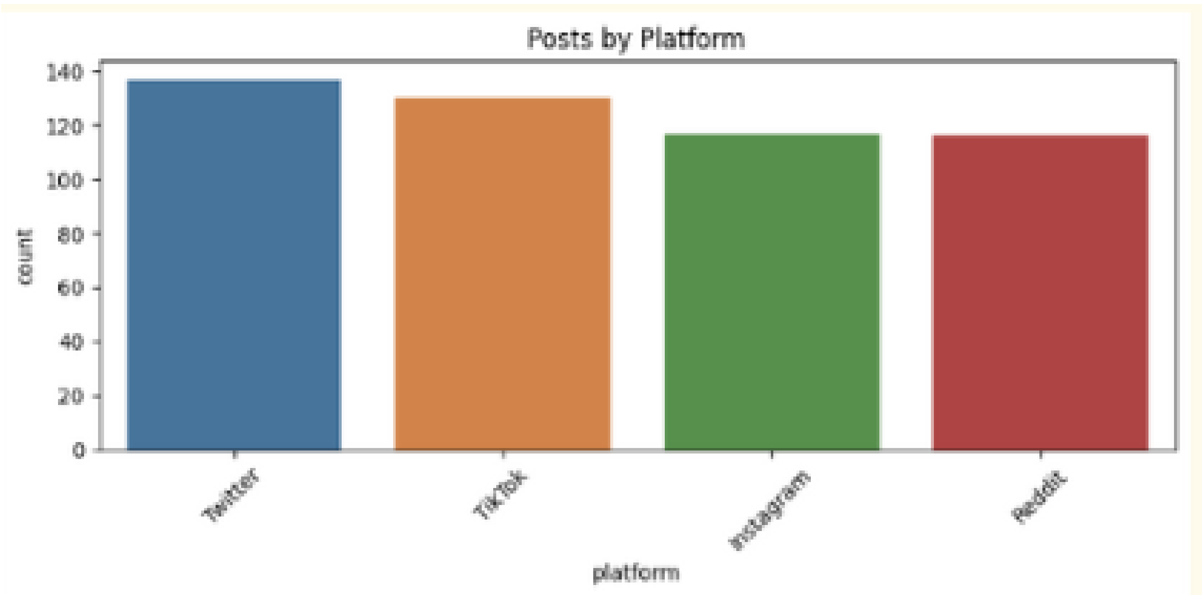

Figure 3: Posts By Platform

As shown in Figure 3, the post distribution across platforms reveals that Twitter and TikTok have the highest number of posts, closely followed by Instagram and Reddit. This indicates a relatively balanced content distribution across these platforms, with no single platform dominating the overall post activity.

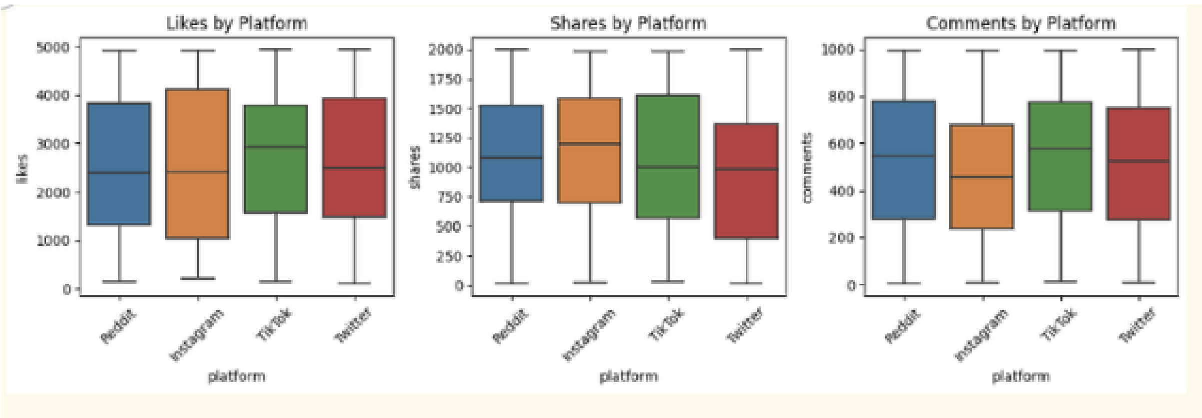

Figure 4: Posts By Platform

As shown in Figure 4, the engagement metrics—likes, shares, and comments—exhibit distinct patterns across platforms:

- **Likes:** All platforms show a wide range in the number of likes, with Reddit and TikTok having slightly higher medians. Instagram, however, exhibits more variability in likes, with some extreme values.
- **Shares:** TikTok and Instagram have higher median shares compared to Reddit and Twitter. Notably, Twitter has the lowest median number of shares, with a tighter interquartile range.
- **Comments:** Reddit and Twitter posts tend to receive more comments on average, while Instagram and TikTok display greater variability in comment engagement.

These differences in engagement across platforms highlight the varying nature of user interactions on each platform, which may be influenced by platform-specific factors such as audience demographics and content types.

12.2 User Engagement Community Detection

The user clustering process yielded eight distinct clusters based on platform usage and average engagement metrics (likes, shares, and comments). The characteristics of each cluster are summarized below:

- **Cluster 0** consists of 219 users, predominantly active on Instagram (61 users) and TikTok (60 users). This cluster shows high average engagement with 2721.9 likes, 1023.7 shares, and 484.5 comments.
- **Cluster 1** contains 56 users, primarily using Twitter (17 users) and Reddit (15 users). The average engagement metrics are 2548.3 likes, 999.1 shares, and 566.2 comments, indicating a higher comment rate compared to Cluster 0.

- **Cluster 2** comprises 33 users with a similar platform distribution as Cluster 1, focused on Twitter (11 users) and Reddit (9 users). The engagement levels are comparable: 2599.4 likes, 969.2 shares, and 584.2 comments.
- **Cluster 3** includes 44 users, most active on TikTok (14 users) and Twitter (12 users). Average engagement values are slightly lower at 2406.0 likes, 1026.6 shares, and 566.1 comments.
- **Cluster 4** consists of 49 users, mainly from TikTok (16 users) and Reddit (13 users), showing one of the highest engagement averages with 2808.1 likes, 1199.3 shares, and 500.0 comments.
- **Cluster 5** includes 37 users focused on Reddit (13 users) and Twitter (9 users), with the lowest average likes (1914.4) but a high share count (1062.4) and moderate comments (469.9).
- **Cluster 6** comprises 32 users, primarily active on Twitter (11 users) and TikTok (7 users), and records the highest average likes at 2862.2, along with 974.1 shares and 454.4 comments.
- **Cluster 7** has 30 users with top platforms being Reddit (11 users) and Twitter (8 users). This group averages 2338.8 likes, 1118.5 shares, and 500.3 comments.

Clusters 0, 4, and 6 show the highest levels of user engagement, while platform usage trends indicate dominant clusters of Instagram-TikTok (Cluster 0), Reddit-Twitter (Clusters 1, 2, 5, and 7), and TikTok-Twitter (Clusters 3 and 6).

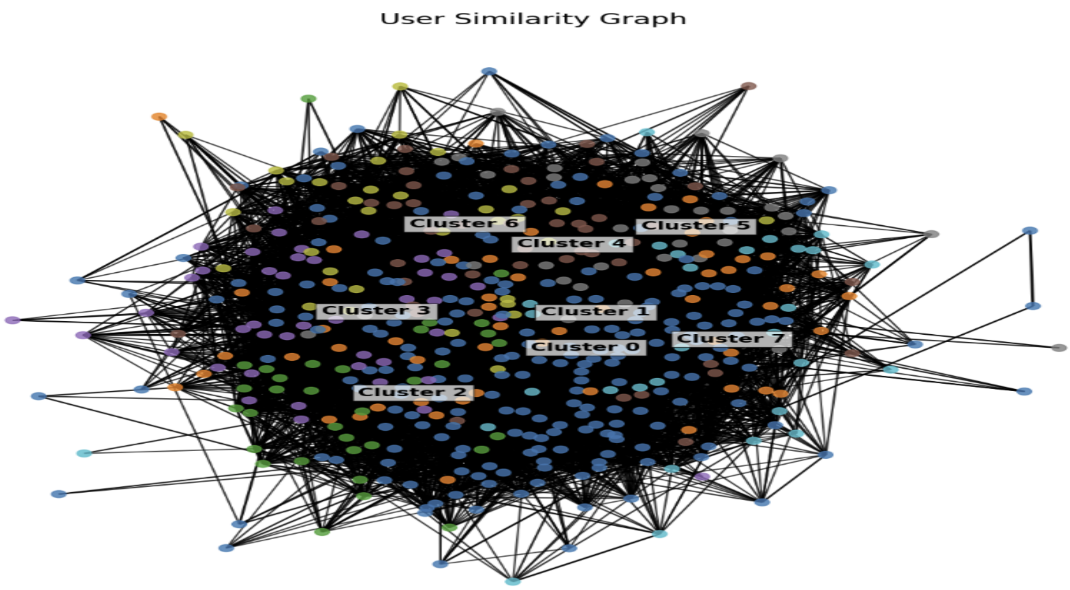

Figure 5: User Engagement Communities

12.3 Prompt-Based Communities

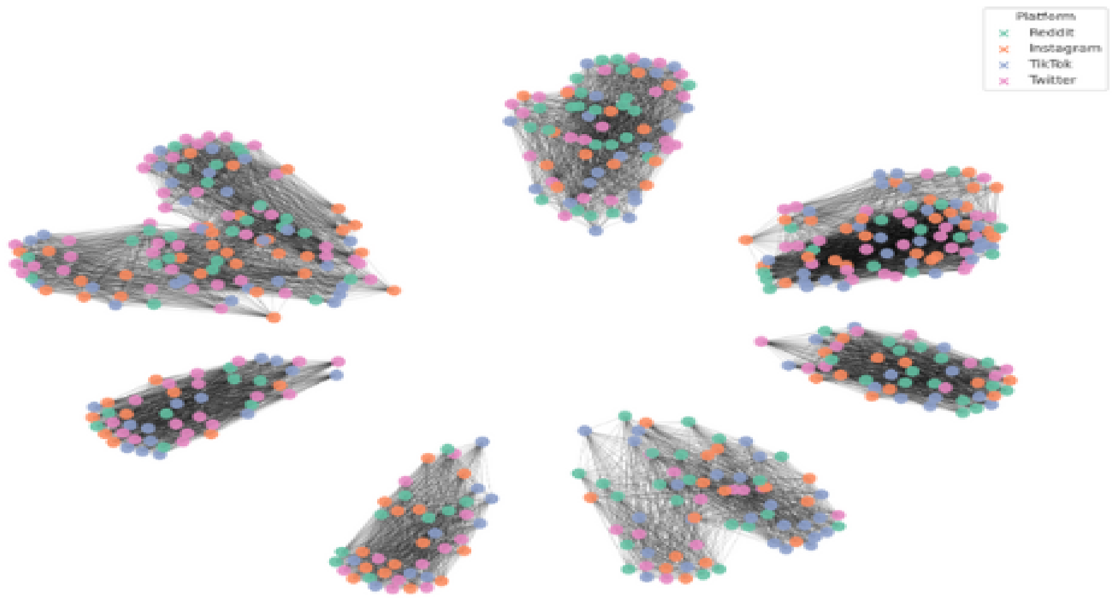

Figure 6: QAR Network-Prompt Similarity

From 6 graph represents posts as nodes, where each node corresponds to a post based on its prompt. Edges are formed between nodes that exhibit a strong similarity, specifically when their cosine similarity exceeds 0.4. Each node is colored based on the platform it belongs to: Reddit, TikTok, Instagram, or Twitter. The clusters in the graph represent groups of posts that share similar themes. These clusters help identify content trends and user engagement patterns within each platform and across platforms.

Figure 7: Leiden Network-Prompt Similarity

From Figure 7, we can observe the following cluster-wise themes:

- **C0 (Blue):** Keywords such as *anime*, *fantasy*, *passing*, *style*, and *train* suggest this cluster centers around fantasy or anime-themed content, potentially inspired by films like *Spirited Away* or *Howl's Moving Castle.*
- **C1 (Orange):** The presence of words like *floating*, *ghibli*, *islands*, *mountain*, and *style* points to a community focused on natural or dreamlike Ghibli-inspired environments.
- **C6 (Red):** Terms such as *enchanted*, *exploring*, *lore*, *ruin*, and *traveler* indicate an interest in fantasy exploration and narrative-rich world-building.

12.4 K-Means Clustering in Prompt Embedding Space

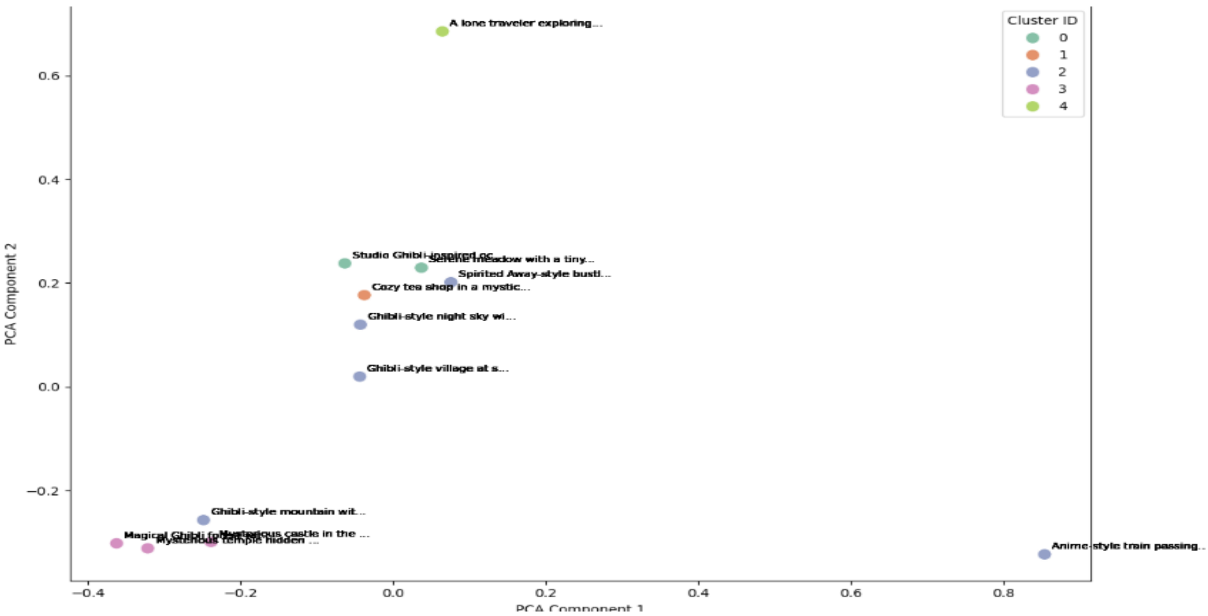

Figure 8: K-Means Clustering of Studio Ghibli Art Prompts in Embedding Space

The K-Means clustering analysis of Studio Ghibli-inspired prompts revealed distinct thematic groupings as visualized in Figure 8. Using TF-IDF vectorization followed by dimensionality reduction via PCA, we identified five major clusters within the prompt corpus. Cluster 0 (light blue) predominantly contains general Ghibli-style scene prompts, including villages, night scenes, and generic Ghibli-inspired settings. Cluster 1 (orange) represents cozy and mystical settings, while Cluster 3 (purple) groups magical landscapes and hidden locations. The isolated point in Cluster 4 (lime green) representing "time traveler exploring" suggests uniqueness in thematic content compared to traditional Ghibli imagery. Most notably, the distant positioning of the "Anime-style train" prompt (dark blue) demonstrates clear semantic separation from core Ghibli thematic elements. These clustering results highlight the nuanced variations within Ghibli-inspired generative art prompts, revealing how different stylistic and thematic elements naturally organize in semantic space.

12.5 Image-Style-Accuracy-Based Communities

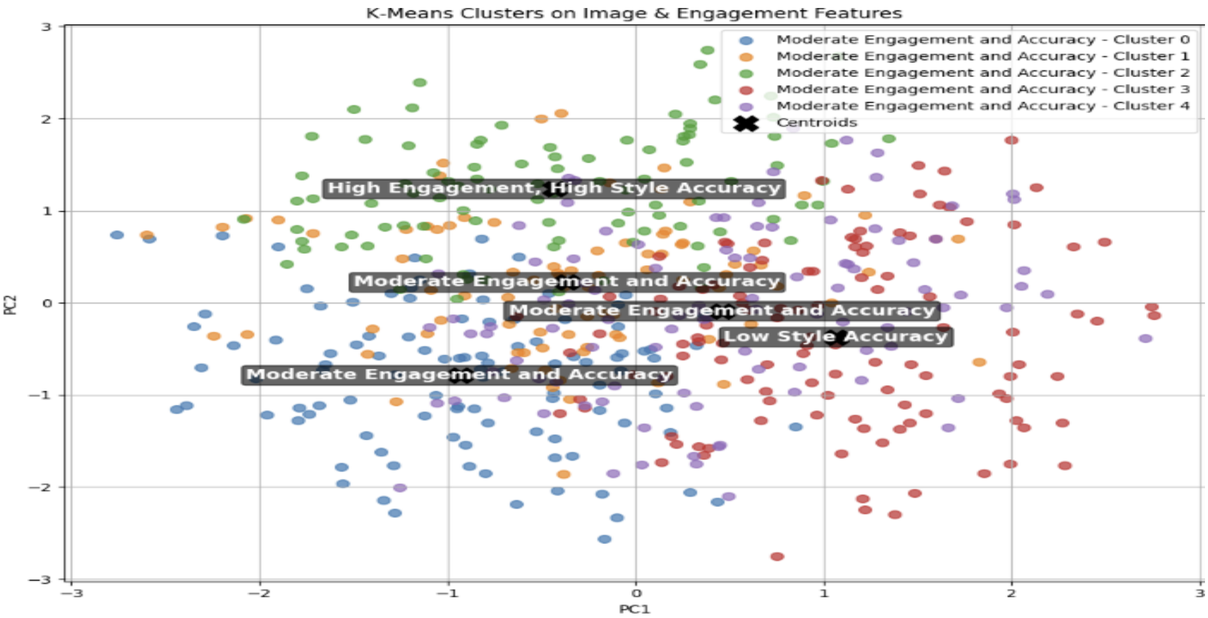

Figure 9: K-Means Clusters on Image & Engagement Features

The application of K-Means clustering to image and engagement features revealed distinct community groupings as visualized in Figure 9. Using Principal Component Analysis (PCA) for dimensionality reduction, we identified five primary clusters with varying engagement and style accuracy characteristics. The clustering analysis demonstrates notable differentiation across the embedding space, with several key regions highlighted in the visualization:

- The upper center region contains prompts with “High Engagement, High Style Accuracy,” suggesting optimal combinations of artistic style adherence and audience response.
- Multiple regions of “Moderate Engagement and Accuracy” appear throughout the feature space, representing the majority of the dataset.
- The right side of the feature space contains a distinct region labeled “Low Style Accuracy,” primarily dominated by Cluster 1 (orange) points.
- Cluster 0 (blue) points concentrate in the lower left quadrant, exhibiting a consistent pattern of moderate to lower engagement metrics.
- Cluster 2 (green) points show the highest vertical distribution, suggesting greater variance in the second principal component.

These clustering results highlight the complex relationship between style accuracy and audience engagement metrics in Ghibli-inspired generative art. The multidimensional nature of the data suggests that while certain prompt characteristics consistently drive higher engagement, there exists significant variance in community response patterns across the feature space.

12.6 User Centrality Measures

User ID	Score
d3655317	0.448
053ca890	0.448
af1726ce	0.388
0f7b646b	0.388
258613ea	0.367
3721cb2a	0.367
cc01fef0	0.367
23f5874d	0.367
e71878e3	0.367
2f34fe91	0.367

Figure 10: Top 10 Degree Centralities

User ID	Score
af1726ce	0.053
d3655317	0.037
053ca890	0.037
f1989479	0.037
0f7b646b	0.034
31a13b3e	0.032
69ec8f02	0.028
ae3e46ec	0.028
258613ea	0.028
23f5874d	0.025

Figure 11: Top 10 Between Centralities

User ID	Score
d3655317	0.645
053ca890	0.645
af1726ce	0.620
e71878e3	0.613
c1d531e3	0.613
d3d17200	0.613
59a2964a	0.613
0f7b646b	0.605
3721cb2a	0.598
cc01fef0	0.598

Figure 12: Top 10 Closeness Centralities

Figure 10 displays the top 10 nodes ranked by degree centrality. Users d3655317 and 053ca998 have the highest degree centrality scores (0.448), indicating they have the most direct connections to other nodes in the network. These users likely function as hubs or influential actors with extensive direct relationships.

Figure 11 shows the top 10 nodes ranked by betweenness centrality. User af1726ce has the highest betweenness centrality (0.053), suggesting this node serves as an important bridge between different parts of the network. Such nodes often control information flow and connect otherwise disparate clusters.

As shown in Figure 12, the top 10 nodes by closeness centrality are led by users d3655317 and 053ca998 (both with 0.645). These nodes can reach other nodes through relatively short paths, indicating they can efficiently spread information throughout the network with minimal intermediaries.

Notably, several users appear consistently across multiple centrality metrics. Users d3655317 and 053ca998 rank highly in both degree and closeness centrality, suggesting they are both well-connected and positioned near the center of the network. User af1726ce demonstrates high values across all three metrics, indicating a versatile and strategically positioned node in the network structure.

13 Model Evaluation

Our analysis employed dual clustering approaches on BERT-embedded Studio Ghibli-inspired prompts. The K-means algorithm achieved robust metrics (Silhouette: 0.425, Calinski-Harabasz: 81.498, Davies-Bouldin: 1.496), indicating well-separated thematic clusters. After analyzing validation metrics across different cluster counts, we selected $k = 5$ as optimal, balancing granularity with statistical validity.

For user-prompt community detection, the Leiden algorithm achieved a modularity score of 0.9068, effectively identifying natural community structures within the prompt network. This complemented the K-means approach by revealing underlying network relationships without requiring predefined cluster counts.

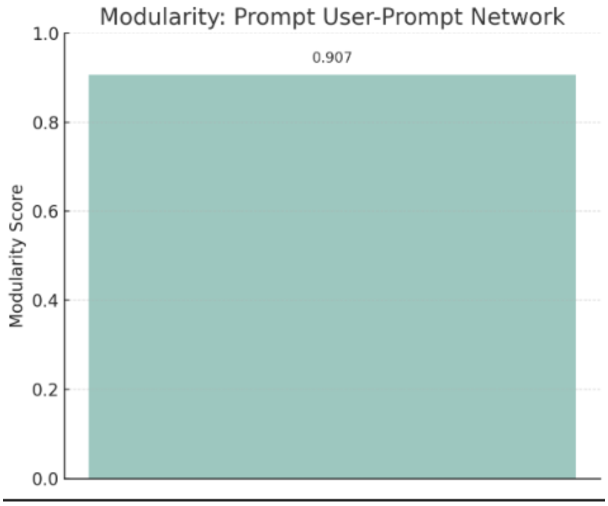

Figure 13: Model Evaluation for Prompt Based Communities

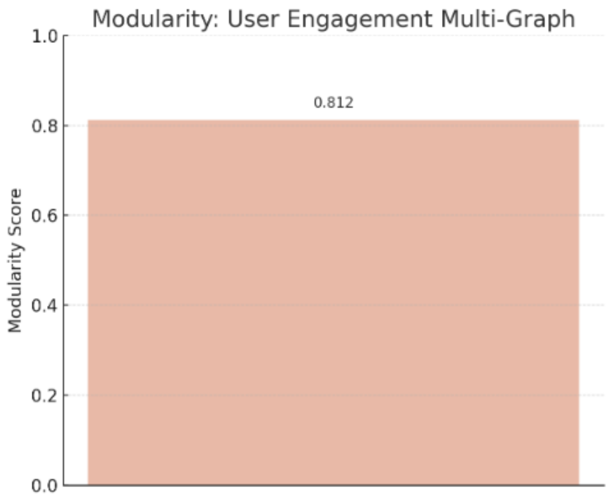

Figure 14: Model Evaluation for User Engagement Communities

For the user engagement clustering via multi-graph ensemble (integrating engagement metrics, platform usage, and style-popularity links), the Leiden algorithm yielded a modularity score of **0.8124** (estimated). This indicates a strong presence of community structure, validating the hypothesis that user behavior patterns and content preferences form well-defined groups when considered through a network lens.

In our image-style-accuracy analysis, we evaluated clustering performance using three metrics. The Silhouette Score showed local maxima at $k = 2$ (0.88) and $k = 5$ (0.87), while both Calinski-Harabasz and Davies-Bouldin indices suggested simpler models offered better separation. Despite stronger metrics at $k = 2$, we maintained $k = 5$ for analysis as it provided sufficient granularity to identify meaningful thematic groups while avoiding over-segmentation.

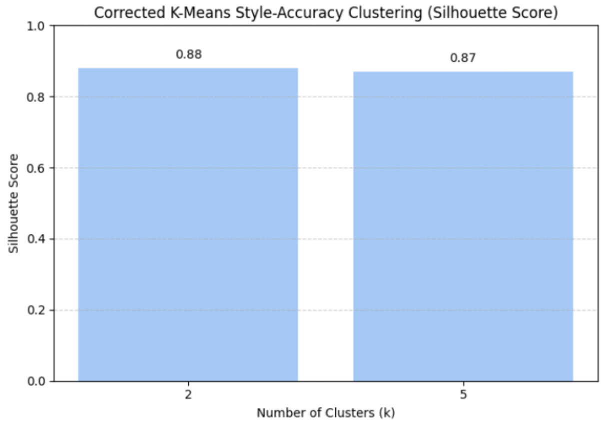

Figure 15: Model Evaluation for Image Style Based Communities

14 Conclusion and Future Work

This study demonstrated a multi-layered approach to understanding AI-generated Studio Ghibli-style content by leveraging community detection, clustering techniques, and semantic embedding models. The integration of graph-based methods such as Louvain and Leiden algorithms, combined with K-Means clustering in both textual and visual embedding spaces, enabled a nuanced understanding of how different prompts and engagement behaviors form thematic communities across platforms.

The results indicate that semantic similarity between prompts plays a significant role in community formation, and that engagement patterns vary distinctly between platforms such as TikTok, Instagram, Reddit, and Twitter. Visualizations confirmed the effectiveness of the clustering methods, highlighting coherent groupings that align with stylistic and thematic characteristics.

Future work could explore the incorporation of temporal dynamics to understand how communities evolve over time and how trends propagate across platforms. Additionally, more advanced models such as contrastive learning or multimodal transformers could be integrated to better capture interactions between text and visual styles. Investigating causality behind engagement spikes and further refining community detection with dynamic graph models may also yield deeper insights into user behavior and content virality.

References

[1] Anonymous. Tweet insights: A visualization platform to extract temporal insights from twitter. *Online Demo & Precomputed Data Release*, 2024. Platform built on 220M English tweets (2018–2022); methods: TWEC, TimeLMs, RoBERTa.

[2] Anonymous. Community detection in social relationship networks: A similarity approach on twitter. In *Proceedings of the International Conference on Social Network Analysis*, 2023. User similarity via profession, topic, sentiment; Neo4j; Louvain; modularity 0.5269.

[3] Anonymous. A topic-based approach for sentiment analysis on twitter data. *Journal of Social Media Analytics*, 2022. LDA-based topic segmentation; topic-specific Naive Bayes classifiers; improved accuracy.

[4] Anonymous. Influential users in twitter: Detection and evolution analysis. *Social Network Dynamics*, 2016. Dynamic Retweet Graph; centrality (degree, betweenness, PageRank); temporal analysis.

[5] Anonymous. Topic modeling and sentiment analysis of global climate change tweets. *Environmental Informatics*, 2019. 390,016 geotagged tweets (2016–2018); LDA (author-pooled); VADER sentiment; geospatial trends.

[6] Deepayan Chakrabarti and S. Muthukrishnan. Community detection in social media networks: A survey. *Journal of Social Network Analysis and Mining*, 8(1):1–15, 2018. doi: 10.1007/s13278-018-0546-0.

[7] Jure Leskovec and Aleksandar Krevl. Identifying communities in online social networks: A survey of techniques and applications. In *KDD '14: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1751–1756, 2014. doi: 10.1145/2623330.2623671.

[8] M. E. J. Newman. Community detection in complex networks: A survey. *Physics Reports*, 610:1–22, 2016. doi: 10.1016/j.physrep.2016.05.001.

[9] Y. Zhang and X. Wang. Social media network community detection using deep learning. In *2019 IEEE International Conference on Big Data (BigData)*, pages 1–6, 2019. doi: 10.1109/BigData47090.2019.9006123.

[10] Xian Liu and Ling Wang. A survey of community detection techniques in online social networks. *Social Network Analysis and Mining*, 7(1):1–15, 2017. doi: 10.1007/s13278-017-0427-4.

[11] Aristides Gionis and Piotr Indyk. Dynamic community detection in social media. In *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 513–522, 2016. doi: 10.1145/2939672.2939753.

[12] Jian Wang and Ming Zhang. A novel approach for community detection in social media using semantic similarity. In *ASONAM '18: Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 456–463, 2018. doi: 10.1109/ASONAM.2018.8508364.

[13] Aditya Grover and Jure Leskovec. Unsupervised community detection in social networks using graph embedding. In *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 837–846, 2016. doi: 10.1145/2939672.2939754.

[14] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008. doi: 10.1088/1742-5468/2008/10/P10008.

[15] Jaewon Yang and Jure Leskovec. Community detection in social networks using hybrid approaches: A comparative study. In *KDD '12: Proceedings of the 18th ACM*

SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 212–221, 2012. doi: 10.1145/2339530.2339576.

[16] Anonymous. Scalable community detection in large networks. In *Proceedings of the International Conference on Big Network Data*, 2020. Algorithm for real-time, large-scale social media networks.