Engineered Homophily in Higher-Order Network Ties: A Case Study of Bluesky Academic Starter Packs

Abstract

In June 2024, Bluesky launched a novel feature called "starter packs," a curated list of up to 150 accounts which a new user can follow with a single click. Unlike the one-by-one follow relationships on older platforms, a starter pack creates many connections at once, making it a higher-order social tie. This study examines how such ties shape homophily, clustering, and community formation within academic networks on the platform. Using a sample of 387 academic starter packs containing 22,026 users and 25,281 memberships, I constructed bipartite and projected networks, ran Louvain community detection, and measured disciplinary and geographic homophily. Three findings stand out: disciplinary homophily at the user level is extreme (H = 0.85); 88.4% of users appear in only one starter pack, sharply limiting opportunities for cross-disciplinary weak-tie bridges in the sense of Granovetter (1973); and geographic homophily is moderate (H = 0.33), suggesting that disciplinary alignment transcends national boundaries more readily than political alignment observed in prior work (Bastos, Mercea, & Baronchelli, 2018). These findings suggest that starter packs operationalise homophily as a platform affordance rather than as an emergent property, which I term *engineered homophily.* Classical homophily (McPherson, Smith-Lovin, & Cook, 2001) emerges slowly from everyday contact in schools, workplaces, and neighbourhoods. Starter packs produce the same outcome in a few minutes of curation. The case study suggests that existing concepts from social network analysis such as homophily, small-world structure, and community formation still apply but need rethinking when platform features bundle many connections into a single curated act. The paper concludes by discussing limitations including sample bias and the absence of a counterfactual follow-network baseline.

1. Introduction

In late 2024, hundreds of thousands of academics - scientists, historians, economists - began leaving X (formerly Twitter) for Bluesky, an online microblogging platform built on the open AT Protocol. This movement was triggered by concerns about platform toxicity under Elon Musk's ownership and accelerated after the November 2024 US election, when journalists, researchers and institutional accounts migrated together in what one *Science* correspondent described as *"the good boring"* (Kupferschmidt, 2025). Bluesky grew from about three million users in early 2024 to more than twenty-six million by year's end (Balduf et al., 2025), and academic discourse on the platform reconstituted itself rapidly. This rapid reconstitution was driven in part by an unusual platform feature. A *starter pack* is a curated list of up to 150 accounts that a new user can follow in a single click (Bluesky, 2024). Starter packs have no direct equivalent on X, Facebook or Instagram. In the language of network science, they are higher-order ties - single connections that link many people at once - operationalised as a platform feature rather than as an emergent structure.

The present study asks a bounded empirical question: within a sample of academic starter packs on Bluesky, how do these higher-order ties structure the social graph they produce? More specifically, does extreme curation (a single user choosing 150 accounts they considers to belong together) generate stronger clustering than classical homophily theory predicts? And do these curated communities reproduce offline geographic divisions in the way that Bastos, Mercea and Baronchelli (2018) found for political echo chambers during Brexit?

To answer these empirically, I scraped 387 academic starter packs from a public directory and enriched the data through the Bluesky API, producing a network of 22,026 unique users linked by 25,281 pack memberships. The methodology combines bipartite network projection, Louvain community detection, eigenvector centrality, and Newman's homophily coefficient, alongside keyword-based inference of disciplinary and geographic attributes. The central finding is that disciplinary homophily in this sample is extreme (H = 0.85 at the user level, against 0.24 under random assignment). 88.4% of users appear in only one pack, mechanically suppressing the weak-tie bridges that Granovetter (1973) identified as carriers of non-redundant information between social clusters. I label this pattern *engineered homophily.* The term is intended to mark a conceptual distinction: classical homophily in McPherson et al. (2001) emerges slowly through repeated contact in bounded foci such as workplaces, neighbourhoods, or schools; engineered homophily is produced instantly by a single curatorial choice and a single click.

2. Literature review

This study sits at the intersection of four bodies of work: the sociology of social network sites, the theory of homophily and weak ties, the spatial analysis of online communities, and a small but growing literature on Bluesky-specific network structures.

2.1 Social network sites and the platform layer: Boyd and Ellison (2007) defined social network sites (SNS) as platforms that let users construct profiles, articulate lists of connections, and traverse those connections. Their key idea was that these platforms mainly make existing relationships visible rather than helping people meet strangers. Bluesky fits this definition but adds something new. Its starter packs allow one person to create a list of many accounts that others can follow instantly. Unlike other platforms like X or Instagram, this shifts some control from individuals to curators who organise entire communities. The work by Boyd and Crawford (2012), along with Tufekci (2014), warned that data from social media is not neutral. Studies using platform data can be biased because of how data is collected (e.g., APIs), who uses the platform, and when the data is gathered. This study acknowledges these biases.

2.2 Homophily and the organic contact assumption: McPherson, Smith-Lovin, and Cook (2001) described homophily as the tendency of people to connect more with similar people. They argued that this happens naturally because people meet others in shared environments. Many later theories build on this idea. Granovetter (1973) showed that weak ties (loose connections) are important because they link different groups and bring new information. Burt (2004) generalised this into the idea of *structural holes*, where people who connect separate groups gain advantages. All three theories share a common assumption: that homophily emerges gradually from organic contact within bounded social settings. Starter packs challenge these ideas because they create connections instantly.

2.3 Community detection, network topology and core-periphery: Newman (2001) applied centrality and clustering measures to scientific co-authorship networks, finding the small-world signature of Watts and Strogatz (1998) and the heavy-tailed degrees of Barabási and Albert (1999). The Louvain algorithm (Blondel, Guillaume, Lambiotte, & Lefebvre, 2008) is now standard for partitioning networks into modular communities. Borgatti and Everett's (2000) *core–periphery* model adds a complementary structural lens: a dense core of highly interconnected nodes coexisting with a sparse periphery. Both apply here. As Section 4 shows, the user-level Bluesky network exhibits extreme modularity ($Q \approx 0.90$) alongside a strong core–periphery signature produced by the high concentration of users in single packs (88.4%).

2.4 The spatial dimension of online networks: Early internet theory predicted that online platforms would dissolve geographic structure. Empirical research has not supported this. Takhteyev, Gruzd and Wellman (2012) showed that X's follow network remained strongly structured by national borders. Dunbar, Arnaboldi, Conti and Passarella (2015) found that online networks mirror offline ones in size and structural properties. Bastos, Mercea and Baronchelli (2018) provide the most direct precedent for the spatial dimension of the present study. Studying the 2016 Brexit referendum on X, they showed that online echo chambers were geographically embedded - partisan clusters mapped onto geographically proximate social networks, particularly on the Leave side. The present study extends this on a different platform, for a different community, and via a different kind of tie (curated rather than organic).

2.5 Starter packs as higher-order ties: A small but growing literature treats starter packs as objects of study. Balduf et al. (2025) analysed the complete population of 335,416 starter packs created in the first half year of the feature's life, showing that they drove a disproportionate share of new follow edges and tended to reinforce existing communities rather than bridge between them. Failla, Citraro, Rossetti and Toschi (2026) released "A Blue Start," a dataset of 39.7 million users and 365,800 starter packs, and framed starter packs explicitly as higher-order social ties. This higher-order framing is taken up here.

3. Methods

3.1 Data collection: I built a two-stage collection pipeline. First, I scraped the public directory blueskystarterpack.com:a third-party aggregator indexing more than 76,000 starter packs to obtain pack references from 26 academic-themed category pages, including academia, academic-research, research, science, computer-science, engineering, mathematics, physics, biology, neuroscience, medicine, artificial-intelligence, machine-learning, universities, institutes, sociology, psychology, economics, history, humanities. Second, for each scraped reference I used the authenticated Bluesky API endpoints `*com.atproto.repo.getRecord*` and `*app.bsky.graph.getList*` to resolve the pack, retrieve its metadata, and fetch current member list. The full collection pipeline is provided in Appendix A.

A two-tier keyword filter was applied to the scraped pack metadata to isolate genuinely academic packs. Packs were retained if the pack name or description contained at least one high-confidence academic identifier (for example, "PhD candidate," "postdoc," "professor," "neuroscientist," "engineering," or any of 80 similar terms) or at least two medium-confidence identifiers (for example, "research" + "science," or "STEM" + "mathematics"). To address the methodological concern that political and activist packs were contaminating an earlier iteration of the sample, I also applied an exclusion list of 45 patterns covering political ideology (*antifa, nationalism, communist, socialist, anarchist, resistance, Trump, MAGA*), diet-focused packs (*UPF, ultra-processed*), and single-issue advocacy (*trans rights, human rights, social justice*). This exclusion list is itself a theoretical choice: it treats the paper as a study of disciplinary homophily, not of the broader politicisation of academic discourse. I retained only packs containing between 5 and 150 members. The upper bound reflects Bluesky's stated cap on starter-pack membership; packs exceeding this are repurposed lists rather than true starter packs. The lower bound excludes trivially small packs. The final sample consists of 387 packs containing 22,026 unique users and 25,281 memberships, after excluding 52 packs on political grounds and 9 for exceeding the size cap. All data are a single cross-sectional snapshot from April 2026.

3.2 Network construction: The raw data form a bipartite graph B = (U ∪ P, E), where U is the set of 22,026 users, P is the set of 387 packs, and (u, p) ∈ E if user u is a member of pack p. From this bipartite graph I constructed two one-mode projections. The *pack–pack projection* connects two packs if they share at least one user, weighted by the number of shared users. The *user–user projection* connects two users if they co-appear in at least one pack, weighted by the number of shared packs. The user–user projection is denser by construction: a single 100-member pack contributes ~5,000 edges to the projection, all of which close triangles.

3.3 Attribute inference: Each pack was assigned a discipline from a nine-category taxonomy: Humanities, Social Sciences, Life Sciences, Physical Sciences, Mathematics/Stats, Computer/AI, Engineering, Methodology/Meta, and General/Unspecified. These were matched against a lexicon compiled from standard disciplinary labels (see Appendix A for the full lexicon). Users inherited the plurality discipline of the packs they appeared in. For users who appeared in packs from multiple disciplines, the most frequent was assigned. For geographic inference, I searched user bios for keywords associated with 21 country groupings (for example, "Boston" → United States). Only 40.7% of users had bios permitting a country inference; homophily measurements using this attribute are restricted to edges where both endpoints were inferable.

3.4 Network metrics: Community detection used the Louvain algorithm with weighted modularity optimisation (Blondel et al., 2008; random seed 42). Homophily was measured using $H = (observed - expected) / (1 - expected)$, where *observed* is the proportion of edges connecting same-attribute nodes and *expected* is the proportion under random assignment proportional to marginal frequencies. H = 1 indicates perfect homophily, 0 random mixing, < 0 heterophily. Transitivity, average clustering, and degree statistics follow standard NetworkX definitions.

4. Results

4.1 Sample composition: The 387 packs were classified into nine disciplines, with Life Sciences (n = 55), Humanities (n = 52) and Social Sciences (n = 42) the largest identifiable groups. Computer/AI (n = 25) and Engineering (n = 17) were substantially represented; 165 packs could not be confidently classified on keywords alone. The user–user projection's giant component contains 21,378 of 22,026 users (97.1%). Geographic inference was possible for 40.7% of users in the giant component. Among those, the United Kingdom (n = 2,970) and the United States (n = 2,221) accounted for 58% of cases, followed by Germany, Canada, Australia and Ireland.

4.2 Network structure: The user–user projection contains 21,378 nodes and 1,176,847 edges (density 0.0049). Node degrees range from 7 to 703 (mean 109.1, median 97), exhibiting a heavy right tail. Transitivity is 0.82 and average local clustering on a 500-node sample is 0.95, both well above the 0.005 expected under an Erdős-Rényi random graph of equal density.

4.3 Community structure: Louvain detection yields 61 communities at the user level (weighted Q = 0.90) and 14 at the pack level (Q = 0.65). Both values are well above Newman's (2006) 0.3 threshold for significant community structure, and the user-level figure (Q = 0.90) is at the upper end of typical values.

4.4 Homophily: For the user–user projection (Figure 2b), the proportion of edges connecting same-discipline users is 0.888, against an expected 0.238 under random assignment, yielding H = 0.85. At the pack level, where each node is classified by its plurality discipline, the figures are observed 0.381, expected 0.241, H = 0.18. Geographic homophily sits between the two at H = 0.33 (observed 0.46, expected 0.20). Figure 2 reproduces a core claim of Bastos et al.

(2018): online clustering is spatially embedded, but at noticeably weaker magnitude than they reported for politically partisan Brexit communication, and in a direction consistent with academic discourse crossing borders more readily than political discourse does.

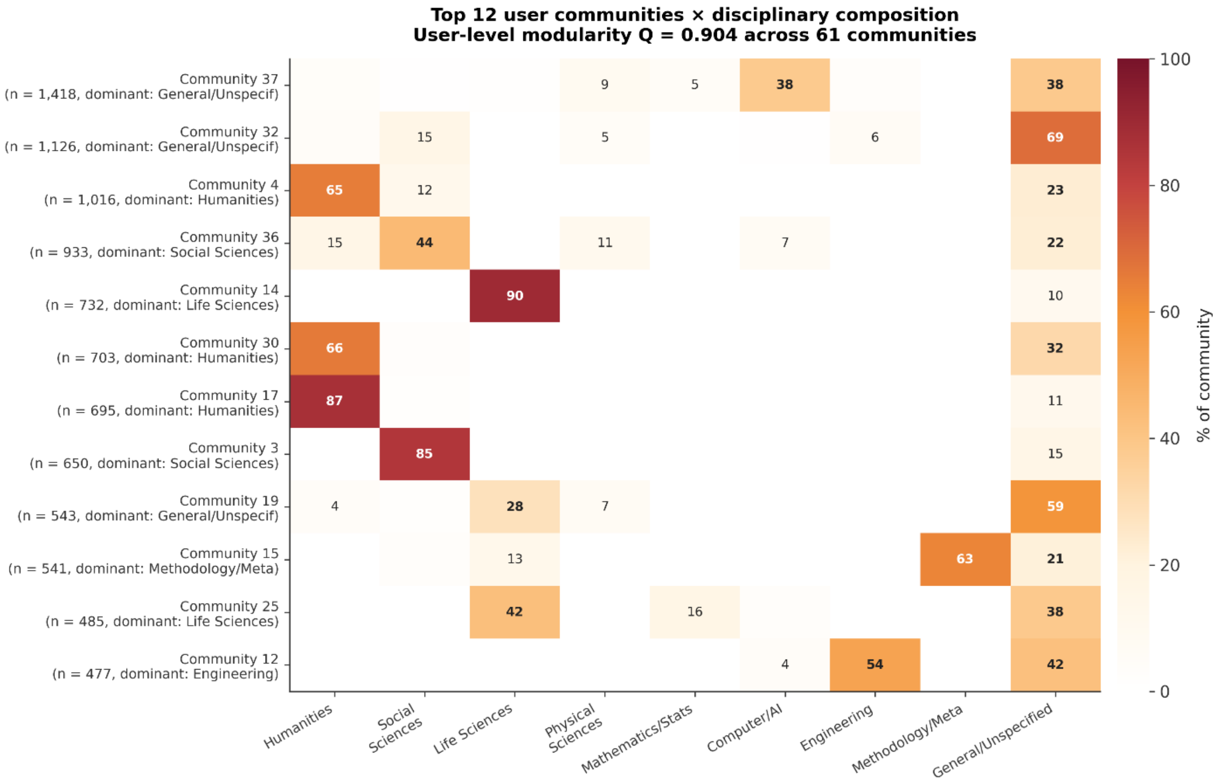

Figure 1. Disciplinary composition of the top 12 user communities recovered by Louvain (user–user projection, weighted modularity Q = 0.90). Each row is a community; each column is a discipline; cells show the percentage of community members in that discipline. Values above 50% indicate near-mono-disciplinary communities (Humanities, Life Sciences, Social Sciences, Engineering).

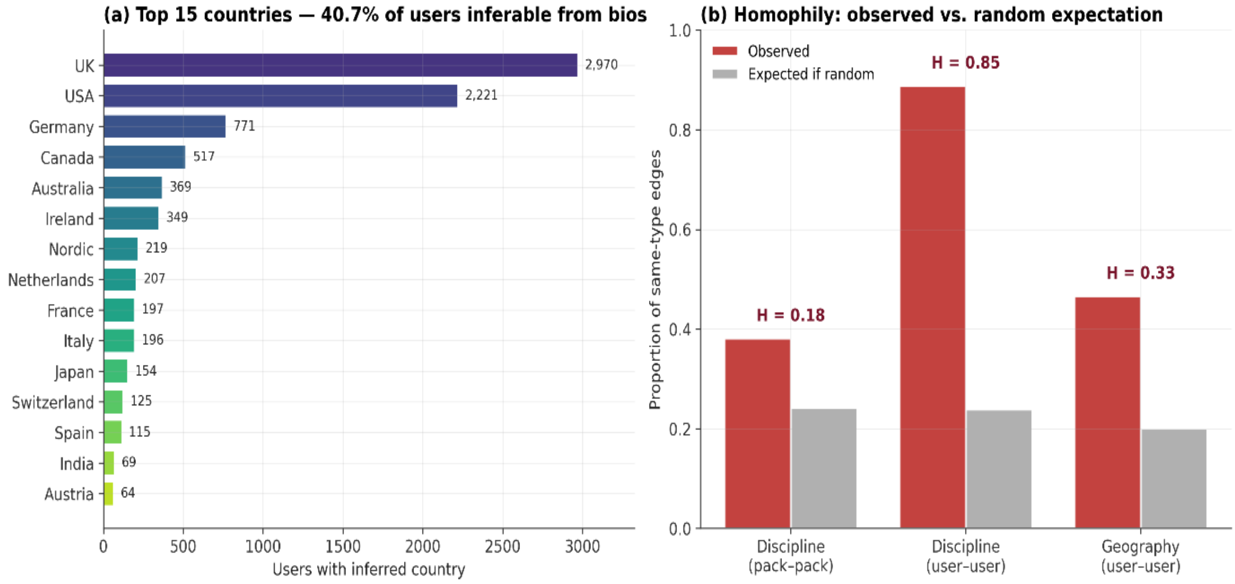

Figure 2. (a) Top 15 countries among users whose bios permitted country inference (40.7% of users in the giant component).

(b) Observed versus expected proportions of same-attribute edges for three attributes. H indicates Newman's homophily coefficient, normalised so that H = 1 is perfect homophily and H = 0 is random mixing.

4.5 The scarcity of bridges: Figure 3 presents the most consequential structural finding. 88.4% of users (19,478 of 22,026) appear in exactly one starter pack. In the Granovetter (1973) framework, cross-cluster bridges are carried by actors with ties in multiple clusters. Starter packs make multi-pack membership rare by construction. The top 25 "bridgers" in the sample, ranked by number of pack memberships, are dominated by institutional accounts such as *Nature, PLOS Biology* and university presses (*Yale, Princeton, Duke, MIT, UC Press*).

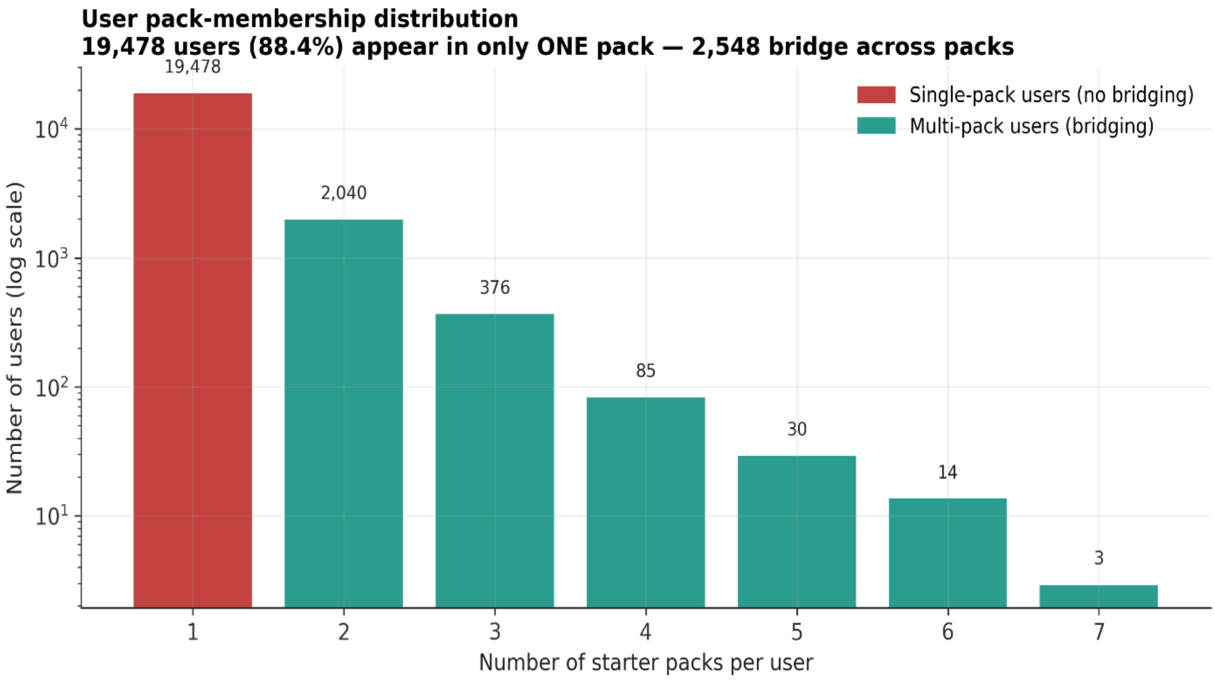

Figure 3. Distribution of pack memberships per user. 19,478 of 22,026 users (88.4%) appear in exactly one starter pack and therefore cannot act as Granovetter-style bridges between packs.

4.6 Pack-level network: Figure 4 visualises the full pack-level network, with nodes positioned by a force-directed layout (ForceAtlas 2; Jacomy, Venturini, Heymann, & Bastian, 2014) and coloured by Louvain community. The result shows clear spatial separation: humanities packs cluster in one region, life sciences in another, computer science and engineering in a third. Even when discipline is assigned by plurality vote on members, packs that share users tend to share a discipline, confirming the user-level homophily reported above.

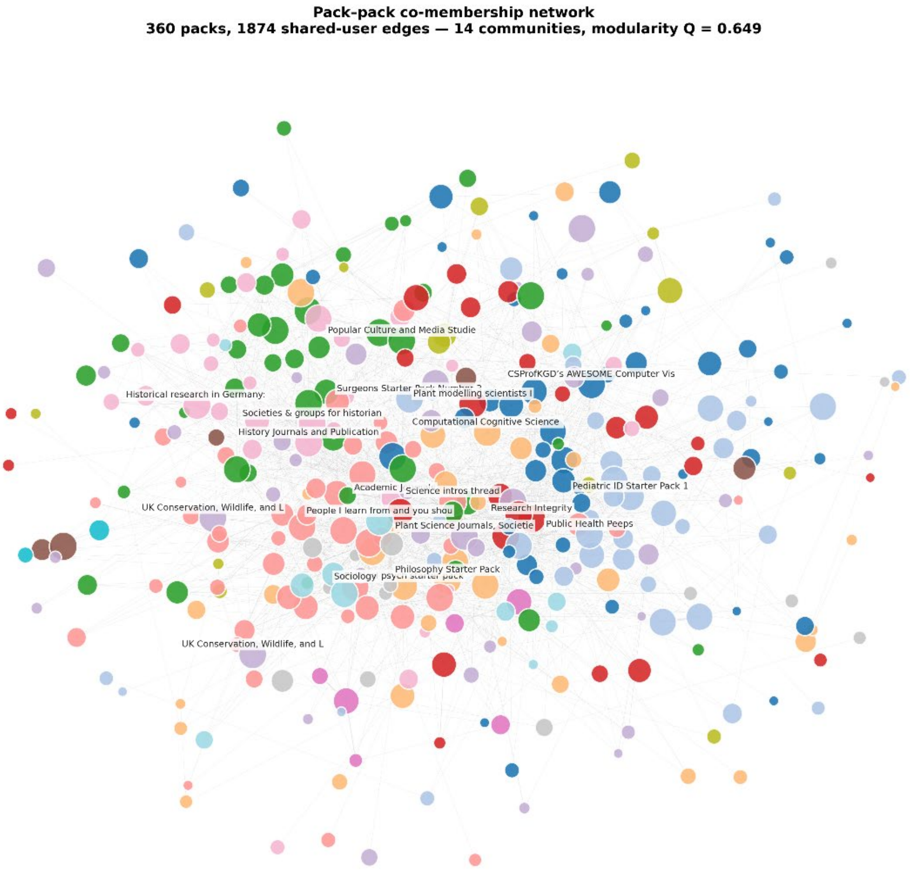

Figure 4. Pack–pack co-membership network. Node size is proportional to membership count; node colour encodes Louvain community assignment. Labels are shown for the 20 largest packs. The visualisation shows spatial separation of disciplinary clusters.

5. Discussion

5.1 Engineered homophily: The empirical centre of this study is the combination of three findings: disciplinary homophily at H = 0.85 at the user level; modular community structure at Q = 0.90; and 88.4% single-pack membership. Read separately, these numbers are unsurprising: a feature designed to bundle similar accounts into curated lists is expected to produce lists of similar accounts. Read together they support a more specific interpretation. Homophily in this sample is not a slow statistical regularity that emerged from years of organic contact; it is the output of a design choice.

A curator spends a few minutes choosing 150 accounts they consider belonging together and at the moment of publication, a higher-order tie exists that connects all of them. Anyone who follows the pack inherits exactly this hyperedge. Homophily is not waited for; it is authored. The outcome resembles classical homophily, but the mechanism that produces it is curatorial rather than emergent, instantaneous rather than accumulated, and located in a single platform act rather than in the structure of everyday social life.

5.2 The weak-tie problem: Granovetter (1973) argued that novel information usually spreads through *weak ties*: connections that link different groups together. This idea assumes that at least some people have links across these group boundaries. Starter packs in this sample do not satisfy this prerequisite. To appear in two packs, a user must be independently noticed and selected by two different curators working from their own attention. For an ordinary user this rarely happens. In the present sample, most users appear in only one starter pack, meaning they have no such cross-group connections. The bridging function on Bluesky, as measured by multi-pack membership, is played almost exclusively by organisations or by well-known individual scholars whose multi-disciplinary visibility predates the starter-pack feature entirely. Ordinary academics do not bridge; institutional accounts do. This has implications for how information diffuses on Bluesky that would be worth investigating in follow-up work.

5.3 Geography: The geographic homophily coefficient of $H = 0.33$ is substantive but clearly smaller than the within-discipline homophily at $H = 0.85$. Reading this alongside Bastos et al. (2018), two interpretations are possible. One is that disciplinary identity in academia travels more easily across borders than political identity does. The other is that the 40.7% inference rate under-powers the geographic measure relative to the discipline measure. Both can be true. In short, academic starter-pack communities are disciplinarily siloed and moderately geographically embedded.

5.4 Relation to classical concepts: Each of the classical concepts invoked in this paper still applies to the Bluesky data, but each is reshaped by the platform layer. Boyd and Ellison's

(2007) definition of an SNS as a site for articulating one's own connections is extended: starter packs let curators articulate *other* people's connections in bundles. McPherson et al.'s (2001) homophily appears at extreme magnitude (H = 0.85) but produced by curatorial design rather than by organic contact. Watts and Strogatz's (1998) small-world structure is present but partly an artefact of the bipartite projection.

6. Limitations

This study uses a convenience sample of 387 starter packs from a single public directory on Bluesky, representing only a subset of the 800+ “Academia” packs on that directory and a very small fraction of the 365,000+ packs platform-wide (Failla et al., 2026), so it should be treated as a case study of visible, English-language academic communities rather than a full census. The sample is also biased toward the UK and USA (about 65%) and relies on English-language keyword filters, which underrepresent non-English academic communities and may affect both network structure and homophily results. The nine-category discipline classification further simplifies complex and interdisciplinary fields, and the use of short pack names and descriptions can lead to misclassification, contributing to a large “General/Unspecified” category and potentially lowering homophily estimates. Additionally, the data were collected at a single point in time (April 2026), meaning the study cannot capture how networks evolve or make causal claims, only structural observations. The lack of comparison with the actual follow network also limits interpretation; future work could examine whether starter packs create or simply reflect homophily. Finally, although all data were publicly available, the study reports only aggregated findings and does not identify individual users except for widely known public or institutional accounts.

Even within these limitations, the case study points to one broader idea worth keeping in mind. Platforms are not just channels for social interaction; they actively shape the ties that form on them. Whether starter packs ultimately help or limit academic communities is a question for future work. What this study makes clear is that the question of who connects to whom must now be paired with who designed the connection.

Word count: 3,326 (excluding title page, references, and appendix)

References

- Balduf, L., Sokoto, S., Ascigil, O., Tyson, G., Castro, I., Baronchelli, A., Pavlou, G., ... Król, M. (2025). Bootstrapping social networks: Lessons from Bluesky starter packs. Proceedings of the International AAAI Conference on Web and Social Media, 19(1), 178–192. doi:10.1609/icwsm.v19i1.35810

- Barabási, A.-L., & Albert, R. (1999). Emergence of scaling in random networks. Science, 286(5439), 509–512. doi:10.1126/science.286.5439.509

- Bastos, M., Mercea, D., & Baronchelli, A. (2018). The geographic embedding of online echo chambers: Evidence from the Brexit campaign. PLOS ONE, 13(11), e0206841. doi:10.1371/journal.pone.0206841

- Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10), P10008. doi:10.1088/1742-5468/2008/10/P10008

- Bluesky. (2024, June 26). Introducing Bluesky starter packs [Blog post]. Retrieved from https://bsky.social/about/blog/06-26-2024-starter-packs

- Borgatti, S. P., & Everett, M. G. (2000). Models of core/periphery structures. Social Networks, 21(4), 375–395. doi:10.1016/S0378-8733(99)00019-2

- boyd, d. m., & Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication, 13(1), 210–230. doi:10.1111/j.1083-6101.2007.00393.x

- boyd, d., & Crawford, K. (2012). Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon. Information, Communication & Society, 15(5), 662–679. doi:10.1080/1369118X.2012.678878

- Burt, R. S. (2004). Structural holes and good ideas. American Journal of Sociology, 110(2), 349–399. doi:10.1086/421787

- Dunbar, R. I. M., Arnaboldi, V., Conti, M., & Passarella, A. (2015). The structure of online social networks mirrors those in the offline world. Social Networks, 43, 39–47. doi:10.1016/j.socnet.2015.04.005

- Failla, A., Citraro, S., Rossetti, G., & Toschi, N. (2026). A Blue Start: A large-scale pairwise and higher-order social network dataset. Scientific Data, 13(1).

- Granovetter, M. S. (1973). The strength of weak ties. American Journal of Sociology, 78(6), 1360–1380. doi:10.1086/225469

- Jacomy, M., Venturini, T., Heymann, S., & Bastian, M. (2014). ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. PLOS ONE, 9(6), e98679. doi:10.1371/journal.pone.0098679

- Kupferschmidt, K. (2025, January 7). As academic Bluesky grows, researchers find strengths — and shortcomings. Science. Retrieved from https://www.science.org/content/article/academic-bluesky-grows-researchers-find-strengths-and-shortcomings

- McPherson, M., Smith-Lovin, L., & Cook, J. M. (2001). Birds of a feather: Homophily in social networks. Annual Review of Sociology, 27(1), 415–444. doi:10.1146/annurev.soc.27.1.415

- Newman, M. E. J. (2001). The structure of scientific collaboration networks. Proceedings of the National Academy of Sciences, 98(2), 404–409. doi:10.1073/pnas.98.2.404

- Newman, M. E. J. (2006). Modularity and community structure in networks. Proceedings of the National Academy of Sciences, 103(23), 8577–8582. doi:10.1073/pnas.0601602103

- Takhteyev, Y., Gruzd, A., & Wellman, B. (2012). Geography of Twitter networks. Social Networks, 34(1), 73–81. doi:10.1016/j.socnet.2011.05.006

- Tufekci, Z. (2014). Big questions for social media big data: Representativeness, validity and other methodological pitfalls. Proceedings of the International AAAI Conference on Web and Social Media, 8(1), 505–514. doi:10.1609/icwsm.v8i1.14517

- Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of "small-world" networks. Nature, 393(6684), 440–442. doi:10.1038/30918

APPENDIX A

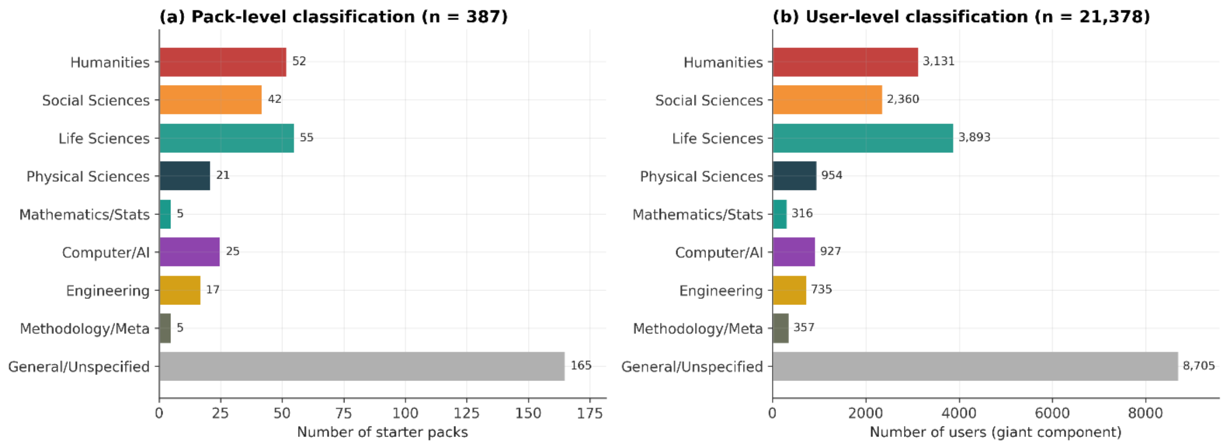

Figure A1. Discipline distribution. (a) Pack-level classification (n = 387). Life Sciences (55), Humanities (52), and Social Sciences (42) are the largest identified categories; 165 packs lacked unambiguous disciplinary keywords. (b) User-level classification (n = 21,378), with discipline inherited from the plurality of packs each user appears in.

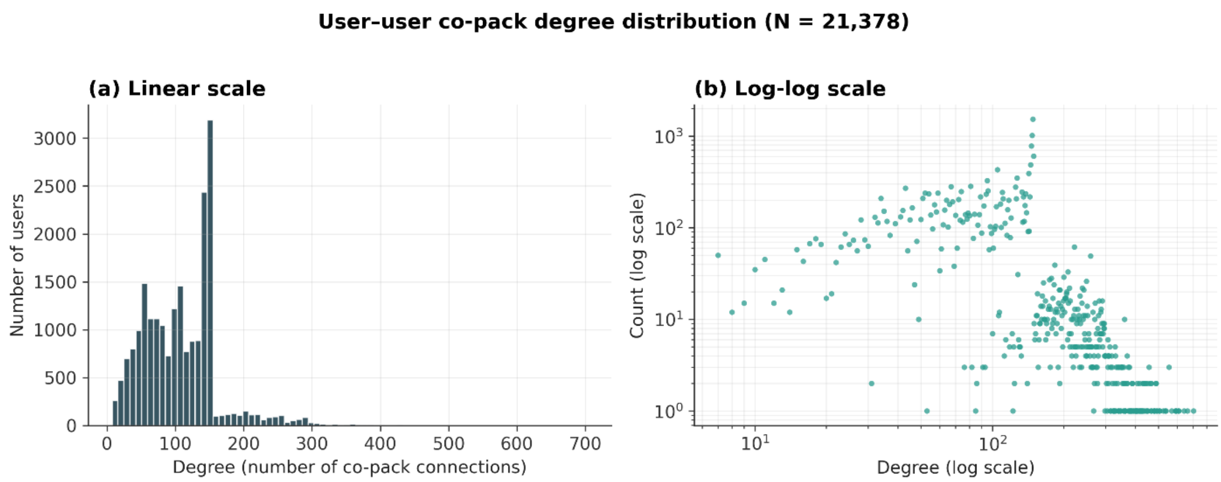

Figure A2. User–user co-pack degree distribution (N = 21,378). (a) Linear scale. The peak near degree 150 reflects Bluesky's 150-member cap on starter packs, since users in a maximum-size pack inherit that many co-pack ties at once. (b) Log-log scale. The distribution has a heavy right tail extending to degree 703.

A.0 Source code:

https://drive.google.com/drive/folders/1SE9kqi2p3AGGQAJM7tDBB7bk6yt2U-Bm?usp=sharing

A.1 Data collection pipeline

The pipeline operates in five stages. First, it scrapes blueskystarterpack.com: a third-party directory indexing more than 76,000 starter packs, across 26 academic-themed category pages, extracting pack references in the form (creator_handle, record_key) by regex. Second, it resolves each unique creator handle to its Decentralized Identifier (DID) using Bluesky's `*com.atproto.identity.resolveHandle*` endpoint. Third, it fetches each pack's record metadata via `*com.atproto.repo.getRecord*`, which yields the pack's name, description, and the URI of its underlying list. Fourth, it applies a two-tier keyword filter (Appendix A.2) to retain only packs whose metadata signals genuine academic content. Fifth, it fetches all members of each surviving pack via `*app.bsky.graph.getList*`. The pipeline applies size filters (5 ≤ members ≤ 150) to exclude both trivially small packs and oversized lists that are not true starter packs. All API calls use authenticated session tokens generated from a Bluesky App Password rather than the user's main account password.

A.2 Keyword lexicons

Three lexicons govern pack inclusion. The strong-academic list (~100 terms) covers career-stage labels (postdoc, professor, tenure-track), profession nouns (sociologist, neuroscientist, engineer), Bluesky academic hashtags (scicomm, econsky, histodons), STEM terms, institutional markers (.ac.uk, .edu, Max Planck), and methodology terms (peer review, preprint). One match in the pack name or description is sufficient. The weak-academic list (~25 generic terms: research, science, STEM, humanities) requires two or more matches, preventing false positives such as "creative writing" or "science fiction." The exclusion list of 45 regex patterns covers political and activist packs (antifa, MAGA, communist, resistance), identity-based packs (trans rights, LGBTQ, decolonial), and dietary or noise packs (UPF, crypto). Excluding identity-based packs is a deliberate methodological choice to isolate disciplinary from identity-based homophily.

A.3 Network analysis pipeline

The full analysis script is provided as `analysis_pipeline.py`. It loads the JSON output of the collection pipeline, builds the bipartite graph $B = (U \cup P, E)$, computes both one-mode projections (user-user and pack-pack, weighted by shared edges in the bipartite graph), runs the Louvain community detection algorithm on each projection (Blondel et al., 2008; random seed 42), computes Newman's homophily coefficient for discipline and geography, and computes eigenvector centrality on the giant component. The script also assigns each pack to one of nine disciplines (Humanities, Social Sciences, Life Sciences, Physical Sciences, Mathematics/Stats, Computer/AI, Engineering, Methodology/Meta, General/Unspecified) by keyword matching and infers each user's country from their bio against a 21-country keyword list.

A.4 Homophily coefficient implementation

The Newman homophily coefficient used throughout this paper is defined as

$H = (\text{observed} - \text{expected}) / (1 - \text{expected})$, where 'observed' is the proportion of edges connecting nodes that share an attribute and 'expected' is the proportion expected under random attribute assignment proportional to the marginal frequency.

A.5 Computational environment

All analysis was performed in Python 3.12 using NetworkX 3.4 for graph construction and centrality, python-louvain 0.16 for community detection, NumPy 2.1 for numerical routines, and Matplotlib 3.9 for figures generated in code. Pack-level network visualisations (Figure 4) were produced in Gephi using the ForceAtlas 2 layout algorithm (Jacomy, Venturini, Heymann, & Bastian, 2014). The random seed was fixed at 42 throughout for both Louvain and any sampled computations.

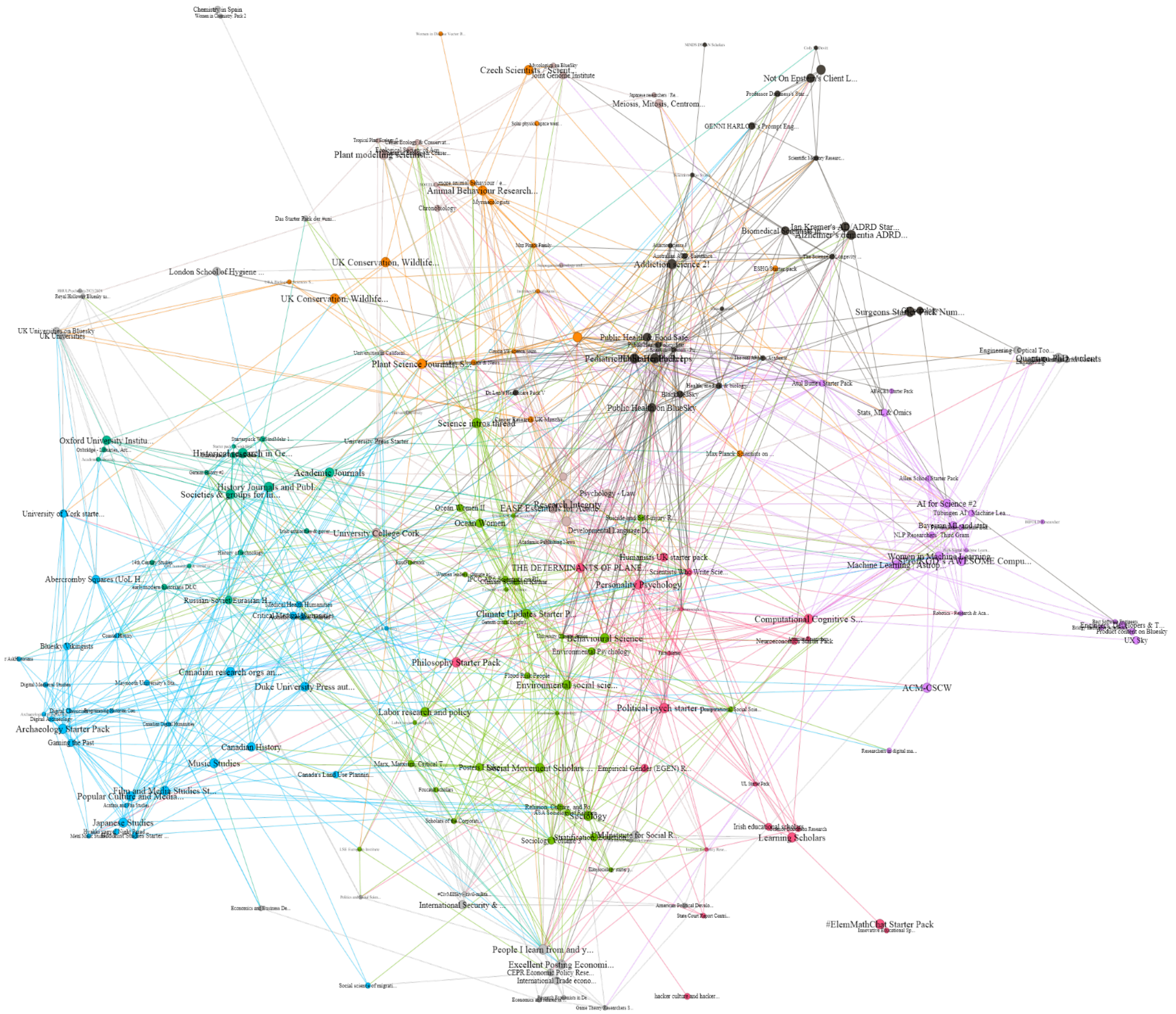

Figure A3. Alternative rendering of the pack–pack co-membership network in Gephi (cf. Figure 4 in main text). ForceAtlas 2 layout produces clearer disciplinary clustering at the cost of label readability..

Python Code

```python
"""
============================================================
Bluesky Academic Starter Pack Collection Pipeline
============================================================
Author: Shyam Joy
Module: IS41510 Social Networks Online and Offline
Submission: April 2026

Purpose
-------
Collect a sample of academic starter packs from Bluesky.
Bluesky's official `searchStarterPacks` endpoint returned
XRPCNotSupported errors throughout 2026 (a known open issue
in the AT Protocol repository), so this pipeline uses a
two-stage approach:

    1. Scrape blueskystarterpack.com (a third-party directory
       indexing 76,000+ starter packs) for pack references.
    2. Resolve each reference through Bluesky's authenticated
       API to obtain the pack's metadata and member list.

Output
------
A single JSON file containing 387 academic starter packs,
22,026 unique users, and 25,281 memberships. See `v2_results.json`
for derived analytical results.


Approximate runtime: 30-60 minutes (rate-limited by both the
directory and the Bluesky API).
"""

import requests
import json
import time
import re
from getpass import getpass


# ----------------------------------------------------------
# AUTHENTICATION
# ----------------------------------------------------------
# Bluesky now requires authentication for all API calls.


BSKY_HANDLE = input("Bluesky handle: ").strip()
BSKY_APP_PASSWORD = getpass("App password: ").strip()

session = requests.post(
    "https://bsky.social/xrpc/com.atproto.server.createSession",
    json={"identifier": BSKY_HANDLE, "password": BSKY_APP_PASSWORD},
    timeout=30,
).json()
HEADERS = {"Authorization": f"Bearer {session['accessJwt']}"}

BASE_URL = "https://public.api.bsky.app/xrpc"
DIRECTORY_BASE = "https://blueskystarterpack.com"


# ----------------------------------------------------------
# STAGE 1 — Scrape the public directory for pack references
# ----------------------------------------------------------
# Each pack is identified by (creator_handle, record_key).


CATEGORY_PAGES = [
    # Core academic
    ("academia", 10), ("academic", 5), ("academic-research", 8),
    ("academics", 5), ("research", 15), ("science", 12),
    ("academic-publishing", 2), ("academic-journals", 2),
    # STEM
    ("computer-science", 5), ("engineering", 5), ("mathematics", 3),
    ("physics", 3), ("biology", 5), ("neuroscience", 3),
    ("medicine", 5), ("chemistry", 3), ("data-science", 3),
    ("artificial-intelligence", 3), ("machine-learning", 3),
    # Institutional
    ("universities", 5), ("institutes", 3),
```

```python
    # Social sciences and humanities
    ("sociology", 3), ("psychology", 3), ("economics", 3),
    ("history", 5), ("humanities", 3),
]

PACK_LINK_RE = re.compile(
    r"/starter-packs/@([a-zA-Z0-9._-]+\.[a-zA-Z0-9._-]+)/([a-z0-9-]+?)-([a-z0-9]{13})"
)


def scrape_category(slug, max_pages):
    """Pull starter-pack references from one directory category page."""
    results, seen = [], set()
    for page in range(1, max_pages + 1):
        url = f"{DIRECTORY_BASE}/{slug}" + (f"?page={page}" if page > 1 else "")
        resp = requests.get(url, timeout=30,
                            headers={"User-Agent": "Mozilla/5.0 research"})
        if resp.status_code != 200:
            break
        new = 0
        for handle, slug_, rkey in PACK_LINK_RE.findall(resp.text):
            key = (handle, rkey)
            if key in seen:
                continue
            seen.add(key)
            new += 1
            results.append({"handle": handle, "rkey": rkey,
                            "pack_slug": slug_, "category_found": slug})
        if new == 0 and page > 1:
            break
        time.sleep(1.0)  # respectful crawl rate
    return results


all_packs = {}
for slug, n_pages in CATEGORY_PAGES:
    for p in scrape_category(slug, n_pages):
        all_packs[(p["handle"], p["rkey"])] = p


# -----------------------------------------------------------
# STAGE 2 — Resolve each handle to its DID
# -----------------------------------------------------------
def resolve_handle(handle):
    """Bluesky internally identifies users by Decentralized
    Identifier (DID). We need this to fetch records."""
    r = requests.get(f"{BASE_URL}/com.atproto.identity.resolveHandle",
                     params={"handle": handle}, headers=HEADERS, timeout=20)
    return r.json().get("did") if r.status_code == 200 else None


unique_handles = {p["handle"] for p in all_packs.values()}
handle_to_did = {h: resolve_handle(h) for h in unique_handles}
handle_to_did = {h: d for h, d in handle_to_did.items() if d}


# -----------------------------------------------------------
# STAGE 3 — Fetch pack record (to get the underlying list URI)
# -----------------------------------------------------------
def get_pack_record(did, rkey):
    r = requests.get(f"{BASE_URL}/com.atproto.repo.getRecord",
                     params={"repo": did,
                             "collection": "app.bsky.graph.starterpack",
                             "rkey": rkey},
                     headers=HEADERS, timeout=20)
    return r.json().get("value", {}) if r.status_code == 200 else None


pack_records = {}
for (handle, rkey), p in all_packs.items():
    if handle not in handle_to_did:
        continue
    rec = get_pack_record(handle_to_did[handle], rkey)
    if rec and rec.get("list"):
        pack_records[(handle, rkey)] = {
            **p,
            "did": handle_to_did[handle],
            "list_uri": rec["list"],
            "name": rec.get("name", p["pack_slug"]),
            "description": rec.get("description", ""),
        }
```

```python
    time.sleep(0.15)


# ----------------------------------------------------------
# STAGE 4 — Two-tier academic filter (see Appendix A.2 for lexicons)
# ----------------------------------------------------------
from filter_lexicons import (
    STRONG_ACADEMIC, WEAK_ACADEMIC, EXCLUDE_PATTERNS,
)


def classify_pack(name, desc):
    """
    Return (is_academic, confidence_level).

    Strict rules:
    - If any EXCLUDE_PATTERNS match, drop the pack.
    - If any STRONG_ACADEMIC term matches, accept (high confidence).
    - If TWO OR MORE WEAK_ACADEMIC terms match, accept (medium).
    - Otherwise drop.
    """
    text = (name + " " + (desc or "")).lower()
    if any(re.search(p, text) for p in EXCLUDE_PATTERNS):
        return False, "excluded"
    if any(k in text for k in STRONG_ACADEMIC):
        return True, "high"
    if sum(1 for k in WEAK_ACADEMIC if k in text) >= 2:
        return True, "medium"
    return False, "low"


academic_packs = {
    k: rec for k, rec in pack_records.items()
    if classify_pack(rec["name"], rec["description"])[0]
}


# ----------------------------------------------------------
# STAGE 5 — Fetch members; apply size filters (5 ≤ n ≤ 150)
# ----------------------------------------------------------
def get_list_members(list_uri):
    members, cursor = [], None
    while True:
        params = {"list": list_uri, "limit": 100}
        if cursor:
            params["cursor"] = cursor
        r = requests.get(f"{BASE_URL}/app.bsky.graph.getList",
                         params=params, headers=HEADERS, timeout=30)
        if r.status_code != 200:
            break
        data = r.json()
        members.extend(data.get("items", []))
        cursor = data.get("cursor")
        if not cursor:
            break
        time.sleep(0.2)
    return members


MIN_MEMBERS, MAX_MEMBERS = 5, 150  # 150 is Bluesky's stated cap

final = {}
for k, rec in academic_packs.items():
    members = get_list_members(rec["list_uri"])
    n = len(members)
    if not (MIN_MEMBERS <= n <= MAX_MEMBERS):
        continue
    final[f"{rec['handle']}/{rec['rkey']}"] = {
        "pack_name": rec["name"],
        "pack_description": rec["description"],
        "creator_handle": rec["handle"],
        "creator_did": rec["did"],
        "category_found": rec["category_found"],
        "members": [
            {"did": m.get("subject", {}).get("did"),
             "handle": m.get("subject", {}).get("handle"),
             "displayName": m.get("subject", {}).get("displayName"),
             "description": m.get("subject", {}).get("description", "")}
            for m in members
        ],
    }
```

```python
    time.sleep(0.25)


# ----------------------------------------------------------
# Save output
# ----------------------------------------------------------
total_memberships = sum(len(v["members"]) for v in final.values())
unique_users = {m["did"] for v in final.values()
                for m in v["members"] if m.get("did")}

output = {
    "packs": final,
    "summary": {
        "n_packs": len(final),
        "n_memberships": total_memberships,
        "n_unique_users": len(unique_users),
    }
}
with open("bluesky_academic_packs_v2.json", "w") as f:
    json.dump(output, f, indent=2)

print(f"Collected {len(final)} packs / {total_memberships} memberships "
      f"/ {len(unique_users)} unique users.")




#==================================================================================================================================
#  Scraped pack data sample output

#   "packs": {
#     "loren325.bsky.social/3mhjifgm4zh2h": {
#       "pack_name": "MAA 2026",
#       "pack_description": "List of people and orgs (who have bsky) connected to the 101st Annual Meeting of the Medieval Academy of A
#       "creator_did": "did:plc:2xnzvumu5d5w5db3kwbl537v",
#       "category_found": "academia",
#       "confidence": "high",
#       "matched_keywords": [
#         "medieval"
#       ],
#       "members": [
#         {
#           "did": "did:plc:w4xrttuwmz6mffcxtkbkgkra",
#           "handle": "medievaljon.bsky.social",
#           "displayName": "Jon Dell Isola",
#           "description": "Medieval Historian. I work on early medieval politics and society. I think about the meaning and purpose of
#         },
#         {
#           "did": "did:plc:a7tcxegrsq6s3q54tnn7pbvp",
#           "handle": "historianofarch.bsky.social",
#           "displayName": "Bonnie Effros",
#           "description": "historian of archaeology, recovering early medievalist, newly implanted Vancouver academic, immigrant and g
#         },
#         {
#           "did": "did:plc:3jdswmwoo3xnuvmbzzouecry",
#           "handle": "hampshirecollege.bsky.social",
#           "displayName": "Hampshire College",
#           "description": "Our students design their own path of study and recruit faculty to guide them and assess work using written
#         },
#         {
#           "did": "did:plc:tsnh7jq3yhmvhh6nd76rzp4h",
#           "handle": "wtfarthistory.bsky.social",
#           "displayName": "Dr. Bryan C. Keene (they/elle/iel)",
#           "description": "Queer nb parent, art historian, curator \ud83e\uddb8\u200d\u2642\ufe0f\ud83d\udd0e\ud83c\udfdb\nProfessor o
own views on art/activism"
#         },
#     }
#     }

#==================================================================================================================================




"""
============================================================
Network Analysis Pipeline
============================================================

The script is deterministic: random seed is fixed at 42 for
Louvain and for sampled clustering coefficient computations.
```

```

Requirements
------------
    pip install networkx python-louvain numpy


"""

import json
import pickle
import time
import random
from collections import Counter, defaultdict
from itertools import combinations

import numpy as np
import networkx as nx
import community as community_louvain

from filter_lexicons import DISCIPLINE_KEYWORDS

random.seed(42)
np.random.seed(42)


# ------------------------------------------------------------
# 1. LOAD DATA
# ------------------------------------------------------------
with open("bluesky_academic_packs_v2.json") as f:
    data = json.load(f)
packs = data["packs"]


# ------------------------------------------------------------
# 2. DISCIPLINE CLASSIFICATION
# ------------------------------------------------------------
def classify_discipline(name, desc):
    """Assign each pack to one of nine disciplinary categories."""
    text = (name + " " + (desc or "")).lower()
    scores = {d: sum(1 for kw in kws if kw in text)
              for d, kws in DISCIPLINE_KEYWORDS.items()}
    if max(scores.values()) == 0:
        return "General/Unspecified"
    return max(scores, key=scores.get)


# ------------------------------------------------------------
# 3. BUILD BIPARTITE NETWORK
# ------------------------------------------------------------
B = nx.Graph()
for pk, p in packs.items():
    B.add_node(pk, bipartite="pack",
               name=p["pack_name"],
               description=p.get("pack_description", "") or "",
               n_members=len(p["members"]),
               discipline=classify_discipline(
                   p["pack_name"], p.get("pack_description")))
for pk, p in packs.items():
    for m in p["members"]:
        if not m.get("did"):
            continue
        u = m["did"]
        if not B.has_node(u):
            B.add_node(u, bipartite="user",
                       handle=m.get("handle") or "",
                       description=m.get("description") or "")
        B.add_edge(pk, u)

pack_nodes = {n for n, d in B.nodes(data=True) if d.get("bipartite") == "pack"}
user_nodes = {n for n, d in B.nodes(data=True) if d.get("bipartite") == "user"}


# ------------------------------------------------------------
# 4. ONE-MODE PROJECTIONS
# ------------------------------------------------------------
# user_to_packs maps each user to the set of packs they belong to
user_to_packs = defaultdict(list)
for pk, p in packs.items():
    for m in p["members"]:
        if m.get("did"):
            user_to_packs[m["did"]].append(pk)
```

```python

# user-user projection (weighted by shared-pack count)
user_edges = defaultdict(int)
for pk in pack_nodes:
    for u, v in combinations(B.neighbors(pk), 2):
        key = (u, v) if u < v else (v, u)
        user_edges[key] += 1

U = nx.Graph()
U.add_nodes_from(user_nodes)
for (u, v), w in user_edges.items():
    U.add_edge(u, v, weight=w)

# Inherit discipline from pack-membership plurality
for u in user_nodes:
    discs = [B.nodes[p]["discipline"] for p in user_to_packs.get(u, [])]
    U.nodes[u]["discipline"] = (
        Counter(discs).most_common(1)[0][0] if discs else "General/Unspecified"
    )
    U.nodes[u]["description"] = B.nodes[u].get("description", "")
    U.nodes[u]["handle"] = B.nodes[u].get("handle", "")

# Restrict to giant component for community / centrality work
giant = max(nx.connected_components(U), key=len)
U_giant = U.subgraph(giant).copy()

# pack-pack projection (weighted by shared-user count)
pack_edges = defaultdict(int)
for ups in user_to_packs.values():
    if len(ups) >= 2:
        for p1, p2 in combinations(ups, 2):
            key = (p1, p2) if p1 < p2 else (p2, p1)
            pack_edges[key] += 1
P = nx.Graph()
for p in pack_nodes:
    P.add_node(p, **B.nodes[p])
for (p1, p2), w in pack_edges.items():
    P.add_edge(p1, p2, weight=w)


# ------------------------------------------------------------
# 5. NEWMAN HOMOPHILY COEFFICIENT
# ------------------------------------------------------------
def homophily(G, attr):

    same, diff = 0, 0
    for u, v in G.edges():
        a = G.nodes[u].get(attr)
        b = G.nodes[v].get(attr)
        if not a or not b:
            continue
        if a == b:
            same += 1
        else:
            diff += 1
    if same + diff == 0:
        return 0.0, 0.0, 0.0
    obs = same / (same + diff)
    freq = Counter(G.nodes[n].get(attr)
                   for n in G.nodes() if G.nodes[n].get(attr))
    total = sum(freq.values())
    exp = sum((c / total) ** 2 for c in freq.values())
    H = (obs - exp) / (1 - exp) if exp < 1 else 0.0
    return obs, exp, H


# Discipline homophily (both projections)
obs_pack, exp_pack, H_pack = homophily(P, "discipline")
obs_user, exp_user, H_user = homophily(U_giant, "discipline")


# ------------------------------------------------------------
# 6. COUNTRY INFERENCE FROM USER BIOS
# ------------------------------------------------------------
COUNTRY_KEYWORDS = {
    "UK": ["uk", "united kingdom", "england", "scotland", "wales",
           "london", "oxford", "cambridge", "edinburgh",
           "manchester", ".ac.uk", "britain", "british"],
    "USA": ["usa", "united states", "america", "american", ".edu",
            "harvard", "stanford", "yale", "princeton", "ucla",
            "berkeley", "mit ", "new york", "boston"],
```

```python
    "Germany": ["germany", "deutsch", "berlin", "munich",
                "max planck", "heidelberg"],
    "France": ["france", "paris", "cnrs", "sorbonne"],
    "Canada": ["canada", "canadian", "toronto", "mcgill"],
    "Australia": ["australia", "australian", "sydney", "melbourne"],
    "Ireland": ["ireland", "irish", "dublin", "ucd", "trinity college"],
    "Netherlands": ["netherlands", "amsterdam", "leiden"],
    # ... 13 further countries (Switzerland, Italy, Japan, Nordics,
    # Spain, Belgium, New Zealand, Austria, Portugal, Brazil,
    # India, China). Full list in repository.
}


def infer_country(bio):
    if not bio:
        return None
    bio_l = " " + bio.lower() + " "
    scores = {c: sum(1 for kw in kws if kw in bio_l)
              for c, kws in COUNTRY_KEYWORDS.items()}
    return max(scores, key=scores.get) if max(scores.values()) > 0 else None


for n in U_giant.nodes():
    U_giant.nodes[n]["country"] = (
        infer_country(U_giant.nodes[n].get("description", "")) or ""
    )
obs_geo, exp_geo, H_geo = homophily(U_giant, "country")


# ------------------------------------------------------------
# 7. LOUVAIN COMMUNITY DETECTION
# ------------------------------------------------------------
P_giant = P.subgraph(max(nx.connected_components(P), key=len)).copy()
part_pack = community_louvain.best_partition(
    P_giant, weight="weight", random_state=42)
Q_pack = community_louvain.modularity(part_pack, P_giant, weight="weight")

part_user = community_louvain.best_partition(
    U_giant, weight="weight", random_state=42)
Q_user = community_louvain.modularity(part_user, U_giant, weight="weight")


# ------------------------------------------------------------
# 8. EIGENVECTOR CENTRALITY
# ------------------------------------------------------------
ec = nx.eigenvector_centrality_numpy(U_giant, weight="weight")
top_ec = sorted(ec.items(), key=lambda x: -x[1])[:25]



# 9. SAVE RESULTS
results = {
    "n_packs": len(pack_nodes),
    "n_users_total": len(user_nodes),
    "n_memberships": B.number_of_edges(),
    "giant_size": U_giant.number_of_nodes(),
    "user_homophily_obs": obs_user,
    "user_homophily_exp": exp_user,
    "user_homophily_idx": H_user,
    "pack_homophily_obs": obs_pack,
    "pack_homophily_idx": H_pack,
    "geo_homophily_obs": obs_geo,
    "geo_homophily_idx": H_geo,
    "user_modularity": Q_user,
    "pack_modularity": Q_pack,
    "user_communities": len(set(part_user.values())),
    "pack_communities": len(set(part_pack.values())),
    "users_in_1_pack": sum(1 for v in user_to_packs.values()
                           if len(v) == 1),
    "users_1pack_frac": (
        sum(1 for v in user_to_packs.values() if len(v) == 1)
        / len(user_to_packs)
    ),
}
with open("v2_results.json", "w") as f:
    json.dump(results, f, indent=2)

print(f"Analysis complete. Q_user={Q_user:.3f}, H_user={H_user:.3f}, "
      f"single-pack frac={results['users_1pack_frac']:.3f}")
```