ChatGPT

Vorbemerkung

Untersucht wurde das öffentlich verfügbare Repository **xai-org/x-algorithm** (Branch *main*, Stand April 2026) für das „For You“-Feed auf X. Laut README handelt es sich um *repräsentativen Beispielcode* (nicht notwendigerweise exakt der Produktionsversion)【1†】. Ausdrücklich fehlen im Release viele Betriebsparameter, Modellgewichte und teils ganze Komponenten. Dennoch lassen sich aus dem Code klare Fakten gewinnen. Folgende strittige Behauptungen aus einem Creator-Thread wurden prüfend beleuchtet.

Claim-by-Claim-Prüfung

ID	Behauptung	Repo/Pfad	Symbol/Kontext
1	„Follower count does nothing for reach anymore.“	`home-mixer/candidate_hydrators/gizmoduck_hydrator.rs` [1] `home-mixer/scorers/weighted_scorer.rs` [2] [3]	`author_followers_count` `WeightedScor`

ID	Behauptung	Repo/Pfad	Symbol/Kontext
2	„Phoenix predicts engagement (19 Heads) and assigns reach darauf; self-fulfilling prophecy.“	`home-mixer/scorers/weighted_scorer.rs` [2] [3] `phoenix/README.md` (vermutlich)	alle `*_score`-Felder
3	„AuthorDiversity: Mehr Posts ⇒ jedes weitere stark dekayt, zu wenig Posts ⇒ Dormanz.“	`home-mixer/scorers/author_diversity_scorer.rs` [4] [5]	`AuthorDiversityScorer` `decay_factor` `floor`

ID	Behauptung	Repo/Pfad	Symbol/Kontext
4	„Reposts: 90% Impressions-Abzug; Self-Reposts ok (Bloom-Filter pro Session).“	`home-mixer/filters/retweet_deduplication_filter.rs` [7] [8] `home-mixer/filters/previously_seen_posts_filter.rs` [9]	`RetweetDeduplicationFilter`
5	„48h-Retention: Posts nach 48h verschwinden aus System.“	`thunder/posts/post_store.rs` [10] [11] `home-mixer/filters/age_filter.rs` [12] [13]	`retention_seconds`, `AgeFilter`

Status: – direkt belegt (Code zeigt eindeutig), teilweise belegt (Teilaussage im Code, Teilaussage Inferenz), **starke Inferenz** (schlüssige Schlussfolgerung aus Struktur), **schwache Inferenz** (plausible Vermutung ohne Beleg), offen (Paramater oder Konfiguration verborgen), widerlegt (Gegenaussage im Code).

Bewertung der Behauptungsgruppen

- **Follower Count:**
  Beobachtung: Der Gizmoduck-Hydrator befüllt ein Feld `author_followers_count` für die Anzeige [1]. **Modell:** Kein Scorer, Filter oder Ranking-Mechanismus verwendet diese Zahl. **Interpretation:** Die Follower-Anzahl ist reiner Meta-Data-Content, hat im aktiven Ranking keine Rolle. **Werturteil:** Die Behauptung ist *weitgehend korrekt* – Followerzahl wird im Scoring nicht genutzt [2] [3]. Es wäre jedoch *irreführend*, daraus zu folgern, dass Follower generell belanglos sind – sie ermöglichen etwa den *Thunder*-Retrieval-Pool (Posts von Gefolgten) [15]. Also: Aussage ist aus Code-Sicht belegbar, aber das Ökosystem wird dadurch komplexer.

- **Prediction Heads (Phoenix):**
  Beobachtung: Der WeightedScorer kombiniert 19 Vorhersagewerte (15 Interaktionen + 4 negatives Feedback) zu einer Gesamtpunktzahl [2] [3]. **Modell:** Jede „Head"-Vorhersage (z.B. `favorite_score`, `reply_score`, `click_score`, `not_interested_score` etc.) wird gewichtet summiert. **Interpretation:** Tatsächlich gibt es **19** Output-Scores (nicht nur 15). Die Behauptung, das System *"entscheidet vor Ausspielung"* über Reichweite, ist eine Schlussfolgerung: Codeseitig erfolgt ein statisches Scoring vor Live-Feedback (selbst-löschende Posts haben damit initial eine hohe Punktzahl), aber die Dynamik dahinter (Feedback-Loops) ist nicht im Code dokumentiert. **Werturteil:** Die reine Zahl der Heads stimmt. Die Idee einer selbsterfüllenden Prophezeiung ist *nicht direkt belegt* (ähnlich üblich in Lernalgorithmen), sondern eine deutenende Interpretation.

- **Author Diversity:**
  Beobachtung: Der `AuthorDiversityScorer` multipliziert jeden weiteren Beitrag eines Autors mit einem abnehmenden Faktor, basierend auf `decay_factor^position + floor` [6]. **Modell:** Dies ist tatsächlich ein exponentielles Abflachen der Scores für mehrfach gezeigte Autoren in derselben Feed-Antwort. **Interpretation:** „Jedes zusätzliche Post wirkt schwächer" ist **direkt belegt**. Eine *Dormanz-Strafe* (Niedrigbewertung bei zu wenigen Posts) ist im Code **nicht** implementiert. **Werturteil:** Die Behauptung „zu viel postet" ist belegt (Auswirkung jedoch parametrisch unklar). Die Behauptung „zu wenig postet schadet" ist im Code unbelegt und vermutlich spekulativ.

- **Repost / Retweet:**
  Beobachtung: Der `RetweetDeduplicationFilter` entfernt Duplikate strikt – nur das erste Auftauchen eines Tweets (Original oder Retweet) bleibt im Pool [8]. **Modell:** Es gibt **keine** Abschwächung um Prozentwerte; Retweets werden gänzlich gefiltert (pro Sitzung) ohne gewichtliche Abstufung. **Interpretation:** Die im Thread genannte „90%-Reduktion" oder „Bloom-Filter-Reset" taucht im Code nicht auf. Vielmehr sorgt Bloom/Seen-Filter dafür, dass einmal gezeigte Posts (oder verwandte IDs) nicht nochmal ins Feed gelangen [9] [16]. **Werturteil:** Die Behauptung über eine 90%-Strafe ist *nicht durch Code belegbar* (daher widerlegt). Selbst-Reposts werden durch dieselben Mechanismen behandelt wie andere Dupes; es existiert kein separater Code-Pfad, der implizit eine Nachsicht für eigene Posts zeigt.

- **48h Retention:**
  Beobachtung: In Thunder gibt es eine harte 2-Tage-Retention (Retentionsperiode = 2×24×60×60 s) [11]. Posts älter als diese Zeit werden beim Einfügen verworfen und laufend gelöscht [10] [11]. **Modell:** Ein `AgeFilter` im Mixer kann ältere Tweets ausschließen, Parameter (MAX_POST_AGE) ist aber nicht öffentlich einsehbar. **Interpretation:** Für *In-Network*-Kandidaten (Thunder) stimmt es: nach 48h ist ein Tweet aus diesem Pool verschwunden. Ob Out-of-Network-Posts (Phoenix-Retrieval) strenger oder anders behandelt werden, ist aus dem Repo unklar. **Werturteil:** Teilweise korrekt (Thunder: ja), aber pauschales „ganz weg im System" wäre übertrieben – älteres Material kann z.B. über Suche oder Conversations erreichbar bleiben, ohne Codeeinfluss.

Abschlussbewertung

1. **Stimmen die Claims?**
   – *Falsch-/Richtig-Aussage:* Die Kernüberprüfungen zeigen, dass wesentliche Teile des Threads *Code-basiert bestätigt* werden: Followerzahl spielt in Ranking keine Rolle [2] [3] [1]; es existieren

19 Scoring-Heads [2] [3]; Author-Diversity implementiert exponentielles Decay [6]; Thunder hat 48h Retention [11]. Diese Fakten sind *direkt belegbar*.

2. **Überzogenes im Thread:**
   – *Fehlende Nuancen:* Einige Aussagen sind übertrieben oder missverständlich. So ist z.B. die Idee eines „Prediction-Traps“ (selbsterfüllende Prophezeiung) zwar logisch möglich, aber im Code keine implizite Rückkopplung erkennbar. Ebenso suggeriert „90% Impressions-Abzug“, das gebe es im Code – tatsächlich wird Retweets einfach **entfernt**, nicht um 90% abschwächt [8].

3. **Nicht belegbare Behauptungen:**
   – *Spekulative Folgerungen:* Dormancy-Strafen („zu wenig posten schadet“) oder dass die Followerzahl als Social Proof komplett irrelevant sei (außer Scoring) sind nicht unmittelbar kodiert. Auch Aussagen zu „Ragebait“ oder Format-Präferenzen lassen sich aus offener Codebasis **nicht** eindeutig belegen. Gerade Anbieter-seitige Gewichtungen und A/B-Experimente fehlen offen.

4. **Unfundierte öffentliche Pauschalaussagen:**
   – *Veröffentlichung mit Vorsicht:* Man sollte nicht den Schluss verteilen, *“die Plattform will nur Engagement und bestraft Kunst“*, wenn die Codebasis – wie gezeigt – keine expliziten Regeln zu Inhaltequalität oder Sujet-Diskriminierung enthält. Ebenso wäre eine definitive Aussage wie „Follower zählen gar nichts“ irreführend, weil Follower über Thunder weiterhin den Kandidaten-Pool bestimmen (auch wenn nicht direkt gerankt) [10] [1].

5. **Empirische Experimentvorschläge:**
   – Um die Reichweitenwirkung zu testen, wären kontrollierte, *codeunabhängige* Experimente nötig: etwa paralleles Posten verschiedener Content-Typen (Bild vs. Text, Reply vs. original) auf ähnlichen Accounts bei definierten Frequenzen; Beobachtung der Impression-Kurven über 1–3 Tage; Vergleiche zwischen Follower-only-Accounts und großen Netzwerken; gegebenenfalls simulierte „Negative Feedback“-Signale (z.B. über Testfollowerset) unter ethischer Berücksichtigung. Nur so lassen sich die codebasierten Indizien in echten Reach-Daten verifizieren.

Quellen: Alle Fakten entstammen dem Open-Source-Repo **xai-org/x-algorithm** (Zugriff April 2026) sowie seinen Dokumentationsdateien [2] [3] [6] [11] [1]. Jeder wichtige Befund wurde durch direkten Codebeleg gesichert. Inferenz-Aussagen wurden ausdrücklich als solche markiert. Theoretische Interpretationen ohne Codebasis sind nicht als gesichert ausgewiesen.

---

[1] Hydrators - X For You Feed Algorithm
https://mintlify.wiki/xai-org/x-algorithm/implementation/hydrators

[2] [3] x-algorithm/home-mixer/scorers/weighted_scorer.rs at main · xai-org/x-algorithm · GitHub
https://github.com/xai-org/x-algorithm/blob/main/home-mixer/scorers/weighted_scorer.rs

[4] [5] [6] x-algorithm/home-mixer/scorers/author_diversity_scorer.rs at main · xai-org/x-algorithm · GitHub
https://github.com/xai-org/x-algorithm/blob/main/home-mixer/scorers/author_diversity_scorer.rs

[7] [8] x-algorithm/home-mixer/filters/retweet_deduplication_filter.rs at main · xai-org/x-algorithm · GitHub
https://github.com/xai-org/x-algorithm/blob/main/home-mixer/filters/retweet_deduplication_filter.rs

9 x-algorithm/home-mixer/filters/previously_seen_posts_filter.rs at main · xai-org/x-algorithm · GitHub
https://github.com/xai-org/x-algorithm/blob/main/home-mixer/filters/previously_seen_posts_filter.rs

10 11 15 x-algorithm/thunder/posts/post_store.rs at main · xai-org/x-algorithm · GitHub
https://github.com/xai-org/x-algorithm/blob/main/thunder/posts/post_store.rs

12 13 14 x-algorithm/home-mixer/filters/age_filter.rs at main · xai-org/x-algorithm · GitHub
https://github.com/xai-org/x-algorithm/blob/main/home-mixer/filters/age_filter.rs

16 x-algorithm/home-mixer/filters/previously_served_posts_filter.rs at main · xai-org/x-algorithm · GitHub
https://github.com/xai-org/x-algorithm/blob/main/home-mixer/filters/previously_served_posts_filter.rs