Beyond Posts: Recommending Conversations in Threads Feed

Xiuyan Ni*
xni@meta.com
Meta Platforms, Inc.
Menlo Park, CA, USA

Qing Feng
qingfeng@meta.com
Meta Platforms, Inc.
Menlo Park, CA, USA

Eric He
eriqqc@meta.com
Meta Platforms, Inc.
New York, NY, USA

Akhil Nadigatla
anadigatla@meta.com
Meta Platforms, Inc.
Menlo Park, CA, USA

Adam Kozdrowicz
akoz@meta.com
Meta Platforms, Inc.
New York, NY, USA

Emerson Wenzel
emersonwenzel@meta.com
Meta Platforms, Inc.
New York, NY, USA

Runbo Jiang
runboj@meta.com
Meta Platforms, Inc.
Menlo Park, CA, USA

Nicholas Joodi
npjoodi@meta.com
Meta Platforms, Inc.
Menlo Park, CA, USA

Xiaoyu Cao
xiaoyucao@meta.com
Meta Platforms, Inc.
Menlo Park, CA, USA

Zichao Li
lizichao@meta.com
Meta Platforms, Inc.
New York, NY, USA

Joy Qiu
crayon@meta.com
Meta Platforms, Inc.
New York, NY, USA

Namrita Perincherry
nperincherry@meta.com
Meta Platforms, Inc.
Menlo Park, CA, USA

Jiangyue Xu
lunajyxu@meta.com
Meta Platforms, Inc.
Menlo Park, CA, USA

Jiaxin Ying
jiaxinying@meta.com
Meta Platforms, Inc.
New York, NY, USA

Feifei Yu
feifeiyu@meta.com
Meta Platforms, Inc.
Menlo Park, CA, USA

Anthony D'Souza
agdsouza@meta.com
Meta Platforms, Inc.
New York, NY, USA

Shuting Wang
shutingwang@meta.com
Meta Platforms, Inc.
Menlo Park, CA, USA

Ivan Ji
ivanji@meta.com
Meta Platforms, Inc.
Menlo Park, CA, USA

Shujian Bu
shujian@meta.com
Meta Platforms, Inc.
New York, NY, USA

Abstract

Nearly all modern social media platforms show only top-level posts in the ranked "For You" Feed; while replies exist, they are hidden into dedicated pages away from Feed. We describe Conversations in Feed ("CI"), the production system powering the recommendations of full post + reply "conversation" chains in the "For You" Feed of Threads, Meta's social media App for conversations. Naively reusing the post-ranking stack for conversations fails: conversations aggregate engagement from both top-level posts and replies leading to significant differences in signal distributions - they skew towards active engagement over passive time-spent - and are subject to higher toxicity and unique content quality risks. Our contributions mainly include three areas across the recommendation stack: (i) a reply-aware ranking objective that keys conversations on the leaf reply, unions engagement signals across the entire reply chain, applies incremental-value debiasing to format-confounded signals such as time spent, and fuses long cross-surface engagement sequences with sparse conversation-keyed features. (ii) an inventory management system that select replies based on a suite of LLM-derived quality signals; and (iii) a Group ID mechanism that suppresses cross-session conversation redundancy via a multiplicative Group-Counter demotion. CIF has been deployed in production on Threads since 2025. An end-to-end deletion test removing all conversations from the ranked feed degraded sessions (daily total

*Corresponding author: Xiuyan Ni.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

In EARL '26 at CIKM '26, November 7–11, 2026, Rome, Italy,

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/XXXXXXX.XXXXXXX

App opens) by −7.21%, impressions (daily number of posts consumed by users) by −7.05%, and time spent (daily total time spent in App) by −5.44% — confirming that conversations meaningfully improve user experience.

CCS Concepts

• **Information systems** → **Recommender systems**; *Retrieval models and ranking*; *Social Networking*; Content ranking.

Keywords

Social Feed Ranking, Conversations, Signal Debiasing, LLM based Inventory Management, Group ID Mechanism

ACM Reference Format:
Xiuyan Ni, Qing Feng, Eric He, Akhil Nadigatla, Adam Kozdrowicz, Emerson Wenzel, Runbo Jiang, Nicholas Joodi, Xiaoyu Cao, Zichao Li, Joy Qiu, Namrita Perincherry, Jiangyue Xu, Jiaxin Ying, Feifei Yu, Anthony D'Souza, Shuting Wang, Ivan Ji, and Shujian Bu. 2026. Beyond Posts: Recommending Conversations in Threads Feed. In *Proceedings of (In EARL '26 at CIKM '26, November 7–11, 2026, Rome, Italy).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

1.1 Why Display Conversations in Feed?

Social-media feeds rank a heterogeneous mix of items — posts, photos, videos — selected from corpora of $O(10^8)$ candidates [3, 7, 11, 17]. Platforms such as Facebook, Instagram, TikTok, and Twitter/X employ multi-stage recommendation pipelines — inventory/pool, retrieval, early-stage ranking (ESR), and late-stage ranking (LSR).

Standard social media feeds only source and rank posts, with the replies being sequestered into a dedicated "comments page" for each post. This works when a product focuses mainly on posts or mainly on replies. However, for threads, Replies drive roughly half of all Threads impressions, with the best replies spread shallowly across many top-level posts. As a result, surfacing the best "conversations" — chains of top-level posts plus replies — *directly* in Feed is critical for Threads (Figure 1).

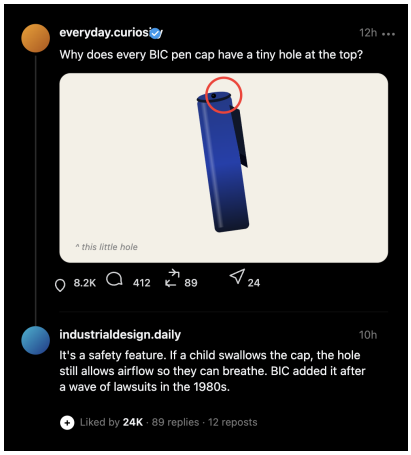


Figure 1: An example of a Threads conversation in Feed. The user marked by the orange dot (`everyday.curios`) authored the *root post* (also called top level post), while the user marked by the blue dot (`industrialdesign.daily`) authored the *reply*. The reply-chain depth is one, that is, a single reply with no nested sub-replies. In this case, root post is the same as parent post. If we have sub-replies, the immediate parent of a reply is itself another reply, whereas the *root post* always refers to the original top-level post (in this example, the post by `everyday.curios`).

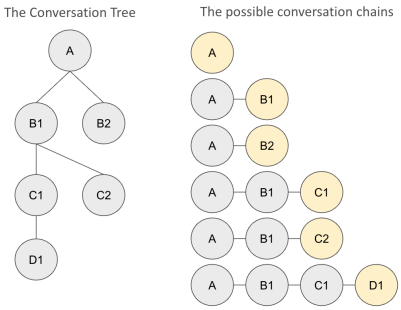


Figure 2: An example of a top-level post and its reply topology (left) and the corresponding conversation chains we could recommend (right). The total number of ranking candidates equals the number of top-level posts + the number of replies, with each reply representing the conversation chain of itself up through its parents to the top-level post, and the top-level post representing the degenerate case of standard post-level ranking.

1.2 Unique Challenges

Today, conversations are sourced and ranked alongside top-level posts, and constitute a meaningful portion of Threads "For You" Feed impressions. The conceptual framework is a simple generalization of the standard ranking problem: instead of showing only the most personalized top-level posts, the goal is to show the most personalized conversation chains. The number of possible conversation chains is isomorphic to the number of top-level posts + the number of replies; see Figure 2 for a visual explanation. While this framing makes conversation recommendation a natural extension of the item recommendation problem, in practice the problem poses fundamental challenges that, to our knowledge, have not been addressed in the literature: **conversations are not posts**. They differ from posts in key ways:

- **Ranking Objective**: The traditional ranking system works by considering engagements on just the post being ranked. Each post is described by a high dimensional feature vector based on the engagements. We translate the likelihood of different engagements into relevancy scores for ranking [12]. However, a conversation has multiple posts that can receive engagement. Conversation ranking must extend the ranking objective to consider the case of multiple-post engagements and integrate this seamlessly into the existing single-post ranking system.
- **Active vs. Passive Engagement**: Reply engagement is predominantly *active reciprocal engagement* - having viewers reply and the root post author liking and replying back - rather than *passive* (time spent scrolling). Time spent-dominated

relevancy scores work well for posts and especially video recommendations, but overvalue low-quality replies that capture attention without generating meaningful interaction.

- **Toxicity Risk**: Conversations have higher toxicity risks. A toxic reply damages the experience of both the viewer and the original poster, creating negative externalities absent from post ranking, yet may result in higher engagement overall. Empirically, we find that integrity penalties must be 8–9× stronger for replies than for posts to maintain quality. More generally, conversations must account for the possibility the user significantly enjoys one part of the conversation (e.g. the reply) while finding other parts of the conversation (e.g. the top-level post) distasteful.
- **Unique User problems**: While posts may only be shown to a user once, ranking conversations carries the risk of showing a user multiple replies to the same post. This duplication problem, which we call "bumping" (Example in Figure 5), is unique to Conversations in Feed and requires special treatments.

These differences make unified ranking of conversations and posts so challenging that nearly all social media Feeds forgo it entirely. To successfully integrate conversations and posts into the ranking stack, we introduce various reply-aware customizations through the ranking stack, yielding significant improvements in both engagement quality and user visitation.

1.3 Our Contributions

We describe Conversations in Feed ("CIF"), the production system that sources conversation chains and ranks them against top-level posts in Threads "For You" Feed. CIF is not a single model: it is a coordinated set of three production components, each accounting for a different layer of the recommendation stack. Our contribution is the engineering case study: we describe how each component was designed, the trade-offs that emerged during deployment, and the online metrics observed in A/B tests on live Threads traffic. The paper is structured around the three components:

(1) **Ranking Objective** (§4.1): CIF extends standard post ranking to conversation chains by keying everything on the leaf reply, taking the union of engagement signals (e.g., likes, time spent) across the reply and its root posts, while applying an incremental-value debiasing step—subtracting a format-aware baseline from implicit signals like time spent — to ensure fair intra-format comparison. Because union scores naturally inflate conversation scores relative to standalone posts, a multiplicative demotion penalty is applied at both ESR and LSR stages. Beyond the union, selective CIF-specific signals (e.g., negative-weighted parent-post-tap for "meta-commentary" replies) capture nuanced engagement dynamics such as "dunk" replies where viewers may block the parent author yet like the reply. On the feature side, conversations are represented by fusing two complementary modalities: long cross-surface, cross-app user engagement sequences that provide deep recent context, and sparse root post keyed features that summarize 60+ days of history where a shared user - item cross tensor lets both modalities attend to each other. Offline evaluation via Normalized Entropy confirms that stacking these sequence and sparse features yields a 1-2% NE improvement across Multi Task Multi Label (MTML) tasks, validating that the two modalities supply complementary signal for conversation ranking.

(2) **Inventory Management** (§4.2): Threads' inventory management system advances millions of daily posts through a directed acyclic graph of quality pools — Cold Start, High Quality, Breaking — where promotion or retirement at each stage is governed by impression targets and engagement signals. For replies, a dedicated hierarchy incorporates engagement from both Feed and Comment Surface, naturally easing the cold-start problem since substantial signal already exists by the time a reply is considered for Feed distribution. Central to our inventory management strategy is a suite of LLM-based quality signals that audit the entire CIF inventory at scale. Grounded in comprehensive user research on how people perceive different types of replies, we fine-tuned large language models to score content along dimensions such as funny, informativeness, and whether a reply is demeaning. These LLM-derived scores are then integrated into pool managements, enabling the system to go beyond raw engagement metrics and surface content that genuinely enhances user experience—while proactively filtering low-quality or harmful replies before they reach broader distribution.

(3) **Group ID Mechanism** (§4.3): We define Group ID as the unique identifier for a set of CIFs that reply to the same Root post. To prevent users from repeatedly seeing CIF in the same group across sessions, for each candidate, we count how many items sharing the same root posts the viewer has already seen, and save the number into a counter keyed on Group ID, we call the number: Group Counter. CIF applies a demotion based on the Group counter when Group Counter ≥ 1. More specifically, it applies a global score penalty, complemented by a quadratic decay of $1/(k+1)^2$ for the k-th item in each conversation group. This mechanism relies on a single new sparse feature Group ID with enrichment placed at post-ESR. Group ID serves dual roles: driving the Group-Counter demotion logic and acting as a sparse input to the later stage MTML model so it can reason about previously seen conversation-tree members. The production launch yielded a dramatic −58% seen-impression rate, together with overall Threads sessions wins, confirming the intervention was correctly targeted at redundancy rather than engagement.

CIF has been deployed on Threads since 2025. To isolate the impact of CIFs, we conducted a deletion test in which all conversation recommendations were removed from the ranked feed. The treatment effect is evaluated along three primary engagement metrics:

- **Sessions**: Total number of App opens per day (One session is one App open).
- **Time Spent**: Total time spent in Threads per day.
- **Impressions**: Total number of posts consumed per day.

The deletion test degrades sessions by −7.21%, overall impressions by −7.05%, and time spent by −5.44%. The engagement metrics



Figure 3: Historic reply preview functionality in other apps

wins demonstrate that conversations are a meaningful part of the user experience.

2 Related Work

Multi-stage feed ranking and Relevancy Scores. Industrial feed-ranking pipelines combine candidate retrieval, early-stage ranking (ESR) and late-stage ranking (LSR) [3, 7, 11]. Multi-task models are used to predict heterogeneous engagement signals [6, 9, 15], which a relevancy score fuses into a scalar objective. The trade-offs of weight setting are subtle [14]; recent work explores RL-based fusion [8]. None of these systems describe *content-type-specific* Value-Model weights deployed for conversations in a unified feed.

Displaying Replies in Feed. Many social media Apps have, at one point or another, displayed replies in Feed (Figure 3). But their approach differs fundamentally from CIF. Rather than sourcing and ranking full conversations and rendering them inline with posts, existing reply preview systems are layered on top of top-level post rankers. For each top-level post, they make a separate, heuristic decision about whether to append a reply preview. This caps their personalization value in a few ways. First, relevance is gated on the top-level post alone, so conversations with great replies but weak top-level posts are rarely shown. Second, per-post reply selection is computationally expensive, so systems fall back on unpersonalized heuristics and lack the personalization power offered by the CIF system.

Reply/Comment ranking. Nearly all apps rank replies in dedicated comment pages. These systems differ from CIF in two ways. First, they rank individual replies, while CIF ranks the entire conversation chain including both the reply and all root posts. Second, they rank replies within a single post, while CIF sources from all conversations and ranks them against each other, as well as against standard top-level posts.

Engagement Toxicity. Milli et al. [10] show that engagement-optimized ranking amplifies divisive content; we also verified this through internal testing. This directly motivates our LLM-based inventory filters.

Debiasing for ranking signals. Time-spent suffer from format and position biases [2, 16]. The standard approach to position bias — IPW or counterfactual learning to rank — assumes a flat-list examination model that does not match nested replies. Our *incremental value* approach (§4.1.2) instead trains a baseline model conditioned on post-format features and subtracts its prediction from the engagement signal; this is closer in spirit to Beutel et al. [1] and Geyik et al. [4] but applied to engagement-quality rather than fairness.

LLM-as-classifier in recommendation. LLM-based content classification is increasingly used to gate or re-rank candidates for integrity reasons [13]. We extend this by deploying an LLM classifier (the *demeaning* filter) at the inventory management stage, where it acts as an offline-batch filter rather than an online re-ranker.

3 System Overview

We present the production system for ranking *Conversations in Feed* (CIF), where selected replies are displayed together with their root posts in the Threads "For You" feed. The feed follows a standard multi-stage ranking architecture: inventory management (corpus construction), retrieval, early-stage ranking (ESR), late-stage ranking (LSR), re-ranking, and final feed assembly, as illustrated in Fig. 4.

Ranking CIF introduces challenges beyond adding a new content type to an existing feed. Replies are substantially more abundant than top-level posts, vary more widely in quality, and derive part of their value from the surrounding conversation context. A CIF candidate is therefore not a standalone item, but a compound unit consisting of a root post and one or more associated replies. The system must decide not only whether a reply is high quality in isolation, but also whether showing that reply adds incremental value relative to showing the root post alone. To address these challenges, we introduce a dedicated reply inventory, group-aware candidate representation, conversation-aware ranking objectives, and listwise controls that jointly govern CIF distribution.

Inventory and Retrieval. Because the reply corpus is much larger than the corpus of top-level posts, directly exposing all replies to the ranking pipeline would create both serving inefficiency and quality risk. We introduce an LLM-driven inventory management layer that continuously evaluates replies before they enter ranking. This layer acts as an agentic quality gate: it identifies low-quality, toxic, or unconstructive replies, prioritizes constructive and contextually relevant responses, and maintains a high-quality candidate pool for downstream retrieval and ranking. Replies are then retrieved through two complementary channels: connected replies from accounts the viewer follows, and unconnected replies from high-quality reply inventory outside the viewer's social graph. This allows the system to surface both socially relevant conversations and broader high-quality discussions while controlling inventory quality and serving cost.

ESR & LSR Ranking. CIF candidates are ranked together with standard top-level posts in a unified multi-stage ranking system, including early-stage ranking (ESR) and late-stage ranking (LSR). Unlike a standard post, a CIF candidate consists of both a root post and an attached reply, so the ranking system must reason about the value of the combined conversation unit rather than the reply in isolation. To support this, we introduce a conversation-level identifier, `Group ID`, which links each reply to its root post

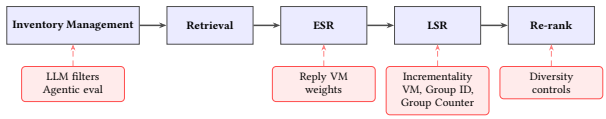


Figure 4: CIF architecture. Blue boxes are the standard four-stage pipeline; red boxes are the conversation-specific components contributed in this paper.

and enables group-level feature construction, modeling, attribution, and downstream controls. Both ESR and LSR models are extended with conversation-aware features derived from this representation, allowing the system to capture signals from the reply, the root post, and the broader conversation context. At the objective level, we design CIF-specific ranking functions to emphasize the incremental value of showing a reply in addition to the root post. This helps mitigate format bias from the richer CIF presentation and enables CIF candidates to be compared more consistently with standard top-level posts in the shared feed ranking stack.

Re-ranking and Diversity Controls. At the re-ranking stage, CIF candidates are blended with standard top-level posts and adjusted by listwise controls to preserve feed quality and diversity. We calibrate overall CIF prevalence so that conversational units complement, rather than dominate, top-level posts. Since multiple replies may share the same root post, we use *Group ID* to prevent repeated exposure to the same conversation thread. Candidates from the same group are progressively demoted when they appear too close together in the ranked list, reducing duplication and clustering. These CIF-specific controls operate alongside standard feed-level rules, including author diversity, topic diversity, and content-type spacing.

4 Integrating CIF into the Ranking System

Our key architectural decision is that conversations compete directly against top-level posts within the same ranking request. However, because conversations are not naturally comparable with top-level posts, the ranking stack has conversation-specific tweaks summarized in Figure 4.

4.1 Extending the ranking objective to conversations

As explained in Figure 2, we can think of conversation ranking as extending the standard problem of ranking the best posts to ranking the best conversation chains, and moreover a conversation chain is uniquely identified by the leaf reply. Thus, in CIF, the reads and writes for all conversation features are done using the leaf reply's identifier, and many parts of the ranking system pretend we are simply sourcing and ranking a reply. This approach presents two operational challenges:

(1) When ranking selects a reply as a winner, the client must render not just the reply but also recursively render the parent posts.
(2) Engagements given to recursive parent posts must be tagged with the reply's identifier so that our label aggregation system can map the engagement back to the source reply.

4.1.1 Taking the Union of Engagement across Reply and Parents. To extend the existing signal set on top-level posts to conversations, we transform each binary signal to be the union of the signal across the reply and all its parents in the chain.

(1) As an example of a binary signal, $p(\text{Like})$ represents the modeled probability the viewer will press the Like button on a post; for conversations, $p(\text{Like}) = p(\text{Like}_{\text{reply}}) \bigcup p(\text{Like}_{\text{parent}})$.
(2) As an example of a float signal, $E[\log(\text{Time})]$ represents the modeled log(time) the viewer spends with a post on the screen. For conversations, $E[\log(\text{Time})] = E[\log(\text{Time}_{\text{reply}} + \text{Time}_{\text{parent}})]$.

The Union transformation allows us to naturally extend all ranking signals on top-level posts to conversations. The weights for the signals remain in sync across conversations and top-level posts across changes to the relevancy scores, which prevents system fragmentation. However, the Union operation presents additional challenges that require addressing (Signal debiasing in §4.1.2).

4.1.2 Signal Debiasing via Incremental Value. Implicit signals such as Time Spent (TS) are the dominant contributors to the LSR relevancy scores. They are also the most format-confounded: conversations simultaneously display both the parent and the reply which shifts time-spent distribution because we union Engagements across Reply and their Parents (§4.1.1). A unified relevancy score would systematically over-promote some formats and under-promote others. We adopt an *incremental value* formulation. For a signal x (e.g., TS and a content item c with format vector $f(c)$ (e.g., is-reply, media-type, we train a baseline model $b(\cdot)$ that learns the average x within each format bucket. The debiased signal is the residual:

$$\tilde{x}(c) = x(c) - b(f(c)). \tag{1}$$

The baseline model is a small Multilayer Perceptron (MLP) recurrently retrained on hourly engagement data; its inputs are exactly the post-format features we wish to control for. The model-based design replaces an earlier hand-bucketed pipeline that did not scale to multiple debiasing dimensions. Once we use $\tilde{x}$ in the relevancy score, intra-format comparison is fair: a reply with above-average time-spent for replies is rewarded, not penalized for being shorter than a typical post. We measure effectiveness offline via Kullback-Leibler (KL) Divergence between per-format score distributions: post-launch, KL shrinks across all reply/quote/media-type buckets, confirming the bias was reduced rather than just hidden by a different score scale.

4.1.3 Conversation Demotion. The score distributions of conversations are the unions of the scores of multiple posts. They will naturally drift higher and exhibit higher variance than the score distributions of standalone posts, and this remains true even after debiasing Time Spent and Like signals. To rebalance the distribution, we apply a multiplicative demotion to the ranking scores of conversations in both the ESR and LSR stages. The specific penalty is empirically determined.

4.1.4 CIF-specific signals. Conversations exhibit diverse and sometimes contrasting signal distributions across the chain:

(1) The user might Block the parent author while Liking the reply. This occurs on "dunk" replies that criticize a controversial parent post; the viewer may dislike the parent post but find the overall exchange entertaining.
(2) Some replies may have low engagement rates, but drastically increase the likelihood of engagement on their parent posts. For example, replies that are "meta-commentary" (e.g. "I broke my knee running to the comments") have low reply-specific engagement rates but get significant distribution because of a high likelihood they get the viewer to tap into the parent post's comments page.

Simply taking the union of parent and reply signals does not give us the flexibility needed to model these engagement combinations. We selectively add "*CIF-specific*" signals that only include the engagement where the engagement dynamics necessitate us to fork. As an example, for "meta-commentary" replies, we include a p(Parent Post Tap) signal with negative weight to discount the oversized impact of the parent post tap likelihood.

4.1.5 Defining Conversations features. The choice of keying conversations on the replies requires us to extend each reply's feature representation to capture context from the entire conversation chain. We do so along two complementary modalities — a user engagement sequence features and a set of sparse, root post keyed features — designed to give the ranking model both broad long-range signal and deep recent context.

In this work, each conversation is represented by fusing two complementary modalities, helping the model learn the value of the full conversation rather than the reply itself. The first is a *long user-engagement sequence* that logs events across all surfaces on which reply engagement occurs — Feed and Comment Sheet Surface (The surface that shows all replies after users tap into a post), and Profile page — and across both Threads and Instagram. This modality provides high-resolution recent context, capturing behavioral patterns conversations depend on, such as how the viewer previously interacted with the root post or with sibling replies in the same conversation tree. The second modality is a set of *sparse, root post keyed features* that summarize 60+ days of user history — counter aggregates rolled up across the full reply chain (e.g., the viewer's prior engagement with sibling replies sharing the same root post, and with root posts whose replies they have previously engaged with). The two views are jointly consumed by the ranking model through a shared user–item cross-tensor: each modality reads from and writes into this tensor, enabling sparse conversation features to serve as queries against the long sequence and vice versa, so that local high-resolution context and long-range root post keyed history can attend to one another rather than being processed in isolation.

4.2 Conversation Inventory Management

Millions of posts are created on Threads each day. Our inventory-management system determines which posts become eligible for distribution in Feed by advancing them through a sequence of *"pools"*, which means sets of candidate media qualified to be recommended. For example, a *Cold Start* pool admits newly created content that has passed integrity and eligibility filters; *High Quality* pools hold content that has accrued hundreds of impressions; and a *Breaking* pool captures posts that have reached thousands of impressions with strong engagement. Formally, the system can be viewed as a directed acyclic graph of quality pools with impression targets: once a candidate reaches its target, engagement-based evaluation criteria determine whether it is promoted or retired.

4.2.1 Standalone Reply Pools. Replies exhibit different characteristics from top-level posts, warranting a dedicated pool hierarchy. Most notably, reply impression targets incorporate views from both Feed and Comment Sheet Surface. Any impressions a reply receives on the comment sheet surface count toward its impression target; likewise, the promotion evaluation criteria aggregate engagement signals from both surfaces. Including reply data yields substantially more signal for assessing reply quality, effectively easing the cold-start problem for CIF: by the time a reply is considered for Feed distribution, engagement signals already exist on both the reply's parent post and the reply's own performance.

4.2.2 LLM-Based Pool Quality Filters. Engagement alone is an insufficient quality signal for replies: demeaning, inflammatory, or polarizing replies may attract engagement yet degrade the content ecosystem over time. This risk is amplified in CIF, where replies are surfaced alongside their parent posts and must be appropriately contextualized. To address this, we fine-tuned an LLM classifier to score replies along multiple quality dimensions including humor, informativeness, and whether the reply is demeaning toward the parent post's author. Replies labeled as demeaning are removed from the inventory pools and become ineligible for any Feed distribution, while the remaining quality signals are used to monitor the healthiness of reply pools. Top-level posts are subject to analogous LLM-based filtering; however, to the best of our knowledge, CIF is the first system in which the *full conversation chain* is provided as context to the classifier, rather than evaluating each post in isolation. This design captures context-dependent toxicity. For example, a reply that appears benign alone but is demeaning when read in the context of its parent post.

4.3 Group ID Mechanism

Across sessions, users frequently see multiple replies whose root post they have already viewed (~7% of replies in feed are directed to previously seen root posts). To mitigate this cross-session "bumping", we apply a Group Counter demotion at final Relevancy scores. Group Counter (GC) is defined based on a single new sparse feature: *Group ID*. Group ID is the identifier of the root post of a reply. Replies that share the same root post belong to the same group.

Storage and lookup. We materialize a server-side mapping from post_id → group_id in a table keyed by post ID. The table is updated when a new reply is created; lookup latency at retrieval/ESR is low single-digit milliseconds.

Use in the system. Group ID is consumed (i) by the Group-Counter demotion logic on top of relevancy scores to prevent cross session "bumping"; (ii) as a sparse feature input to the LSR MTML model, where it allows the model to reason about which other media in the same conversation tree the viewer has seen; (iii) diversity controls to prevent repeated exposure to the same conversation thread by progressively demoting replies with the same Group ID in the same session.

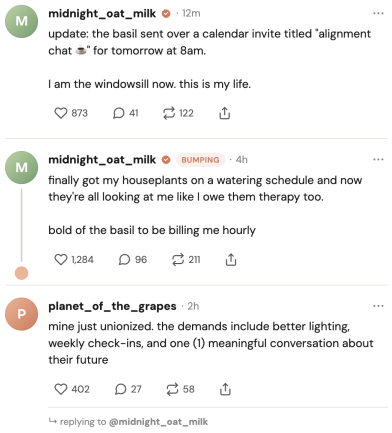


Figure 5: An example of reply bumping — seeing the same top-level post multiple times due to a reply.

For each candidate c_k at session t, we compute GC, the number of media items with the same Group ID (i.e., sharing the same root post) that the viewer has previously seen. When GC ≥ 1 we apply a global score demotion using a complementary quadratic decay:

$$V_{\text{dedup}}(u, c_k) = V(u, c_k) \cdot \big(1/(k{+}1)\big)^2, \tag{2}$$

where k is the 0-indexed position of c_k within its conversation group. Rollout impact is reported in Section 5.

5 Results

All results are from online A/B tests on live Threads traffic using Sessions, Time Spent and Impressions.

5.1 End-to-End Deletion Test

We maintained a long-term deletion test in which we disabled all CIFs. The deletion test degrades sessions −7.21%, overall impressions −7.05%, time spent −5.44% (listed in Table 1). The empirical basis for our investment in end-to-end conversations recommendations.

Table 1: Overall. All deltas are No CIFs vs. Production

Component	Sessions	Impr.	Time Spent
Overall	−7.21%	−7.05%	−5.44%

5.2 Per-Component Impact

5.2.1 Feature & Model Improvements.

Offline evaluation via Normalized Entropy (NE).. We measure the impact of conversations features and model improvements (§4.1.5) with Normalized Entropy [5] on every MTML head independently. For a binary task with label $y_i \in \{0, 1\}$, predicted probability p_i, and empirical positive rate $\bar{p} = \frac{1}{N}\sum_i y_i$ over N examples, NE is the binary cross-entropy of the model normalized by the entropy of the constant-$\bar{p}$ predictor:

$$\text{NE} = \frac{-\frac{1}{N}\sum_{i=1}^{N}\big[\, y_i \log p_i + (1-y_i)\log(1-p_i)\,\big]}{-\big[\,\bar{p}\log\bar{p} + (1-\bar{p})\log(1-\bar{p})\,\big]}. \tag{3}$$

NE is invariant to the marginal label rate, making it directly comparable across heterogeneous heads (e.g., p(Time Spent) vs. p(Like)) and across data slices with different positive rates; lower NE indicates a better-calibrated model. Stacking the sequence features and the conversation-specific sparse features described above yields a 1–2% *NE improvement* across the LSR MTML tasks (skip, time spent, like, reply, etc.), confirming that the sequence and sparse modalities supply complementary, non-redundant signal for conversation ranking.

Online Evaluation. We also evaluated through A/B tests on conversations features.

Table 2: Online results for Conversations Features. All deltas are A/B pretest vs. control

Component	Sessions	Impr.	Time Spent
Conversations Features (§4.1.5)	+2.8%	+4.2%	+2.5%

5.2.2 LLM-Based Quality Filters. The demeaning-reply classifier (§4.2.2) operates as a batch filter at the inventory stage, removing flagged replies before they enter the retrieval candidate pool. The filter is engagement-neutral by design — it targets content that attracts engagement *despite* degrading the ecosystem — and achieves its intended effect on content quality with no statistically significant regression in engagement metrics. Approximately 17% of replies are classified as "demeaning" by the LLM. The impact of inventory managements can not be measured independently, and are reflected in the deletion test together with other improvements in Table 1. The Demeaning Reply filter is provided by Llama Model and fined tuned using Gemini 2.5. We evaluated the model performance on a human labeled golden set.

Table 3: LLM based Classifier: Demeaning Reply

Signal	Precision	Recall
Demeaning (§4.2.2)	70.4%	46.3%

5.2.3 Signal Debiasing. Before debiasing, CIF candidates traversed the ranking funnel at disproportionately high rates relative to top-level posts, resulting in a sub-optimal allocation for conversation recommendations. We therefore apply incremental-value debiasing to the time-spent family of signals and evaluate the intervention through online A/B testing. The aggregated effects are reported in Table 4.

Table 4: Signal Debiasing. All deltas are A/B pretest vs. control

Component	Sessions	Impr.	Time Spent
Signal Debiasing (§4.1.2)	+1.79%	+1.86%	+4.06%

5.2.4 Cross-Session Redundancy Reduction. Prior to the Group ID mechanism, ∼7% of conversation impressions involved a parent post the viewer had previously seen — users were repeatedly shown the same top-level post with different replies across sessions. The Group Counter demotion (§4.3) reduced this *previously seen impression rate* (PSIR) by −58%. Engagement metrics improved alongside the quality improvement, confirming that the redundancy was displacing novel content rather than serving a useful reinforcement function. The online A/B test results are listed in Table 5.

Table 5: Group ID mechanism. All deltas are A/B pretest vs. control

Component	Sessions	Impr.	Time Spent
Group ID Mechanism (§4.3)	+0.37%	+1.14%	+0.53%

6 Conclusion

Conversations in Feed (CIF) is the production reply-recommendation system at Threads. Its core engineering thesis — that replies are not posts, and that ranking, retrieval, modeling, and inventory management must each carry reply-specific structure — is supported by both a long-term holdout (sessions −7.21%, Time Spent −5.44% when replies are removed) and per-component A/B launches in 2025. The components — LLM-gated inventory management with agentic offline evaluation, dedicated reply retrieval, incrementality-based debiasing combined with reply-specific Value-Model weights and Group-Counter cross-session demotion, and the Group ID modeling feature — are jointly necessary. Open directions include (i) automated bias-feature detection beyond implicit signals, (ii) personalization of reply Value-Model weights.

References

[1] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H. Chi, and Cristos Goodrow. 2019. Fairness in Recommendation Ranking through Pairwise Comparisons. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 2212–2220.

[2] Jia Chen, Jiaxin Mao, Yiqun Liu, et al. 2024. Unbiased Learning to Rank Meets Reality: Lessons from Baidu's Large-Scale Search Dataset. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 1691–1700.

[3] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys)*. 191–198.

[4] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 2019. Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 2221–2231.

[5] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, and Joaquin Qui nonero Candela. 2014. Practical Lessons from Predicting Clicks on Ads at Facebook. In *Proceedings of the 8th International Workshop on Data Mining for Online Advertising (ADKDD)*. 1–9.

[6] Jiacheng Li, Huicai Zhong, et al. 2024. HoME: Hierarchy of Multi-Gate Experts for Multi-Task Learning at Kuaishou. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 5313–5322.

[7] Weiwen Liu et al. 2022. Neural Re-ranking in Multi-stage Recommender Systems: A Review. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*. 5479–5486.

[8] Ziru Liu et al. 2025. xMTF: A Formula-Free Model for Reinforcement-Learning-Based Multi-Task Fusion. *arXiv preprint arXiv:2504.05669* (2025).

[9] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. 2018. Modeling Task Relationships in Multi-task Learning with Multi-gate Mixture-of-Experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 1930–1939.

[10] Smitha Milli, Luca Belli, and Jeffrey T. Hancock. 2025. Engagement, User Satisfaction, and the Amplification of Divisive Content on Social Media. *PNAS Nexus* 4, 1 (2025), pgae537.

[11] Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, et al. 2019. Deep Learning Recommendation Model for Personalization and Recommendation Systems. *arXiv preprint arXiv:1906.00091* (2019).

[12] Xiuyan Ni, Yang Yu, Peng Wu, Youlin Li, Shaoliang Nie, Qichao Que, and Chao Chen. 2019. Feature Selection for Facebook Feed Ranking System via a Group-Sparsity-Regularized Training Algorithm. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*. 2085–2088.

[13] Claire Jo Magdalena Wojcieszak Anshuman Chhabra Rajvardhan Oak, Muhammad Haroon. 2025. Re-ranking Using Large Language Models for Mitigating Exposure to Harmful Content on Social Media. *arXiv preprint arXiv:2501.13977* (2025).

[14] Barry Smyth, Lucas Maystre, and Mounia Lalmas. 2023. Choosing the Right Weights: Balancing Value, Strategy, and Noise in Recommender Systems. *arXiv preprint arXiv:2305.17428* (2023).

[15] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive Layered Extraction (PLE): A Novel Multi-Task Learning Model for Personalized Recommendations. In *Proceedings of the 14th ACM Conference on Recommender Systems (RecSys)*. 269–278.

[16] Fangyu Wang et al. 2025. Debias Can be Unreliable: Mitigating Bias Issue in Evaluating Debiasing Recommendation. In *Companion Proceedings of the ACM Web Conference 2025 (WWW)*.

[17] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Anurag Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending What Video to Watch Next: A Multitask Ranking System. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys)*. 43–51.