UDC: 004.93 https://doi.org/10.15407/jai2026.02.127

O. Ostrovska[1], V. Lyashkevych[2]
[1,2] Ivan Franko National University of Lviv, Ukraine
50, Drahomanova str., Lviv, 79005
[1] oksana.ostrovska@lnu.edu.ua
[2] vasyl.liashkevych@lnu.edu.ua
[1] https://orcid.org/0009-0006-3376-8448
[2] https://orcid.org/0000-0003-2810-6061

HOW UKRAINE READS THE WAR: A FIVE-YEAR DATASET OF 20 TELEGRAM NEWS CHANNELS AND WHAT THEIR ENGAGEMENT PATTERNS REVEAL

Abstract. Telegram has emerged as the dominant information platform in Ukraine, where many users rely on it daily as their primary news source, emergency alert system, and government communication channel. Despite this central role in one of the most significant armed conflicts of the twenty-first century, large-scale longitudinal datasets capturing the dynamics of Ukrainian Telegram channels remain largely absent from the research literature, leaving a critical gap in the study of conflict-zone information ecosystems. This paper presents a comprehensive dataset collected from 20 Ukrainian Telegram channels spanning January 2021 to March 2026 — a period encompassing the pre-invasion baseline, the acute shock of Russia's full-scale invasion on February 24, 2022, and the prolonged attritional phase that continues at the time of writing. The channels are distributed across six editorially diverse categories — Official Government, Military, Mainstream Media, Independent Journalist, Anonymous/Aggregator, and Regional — and the dataset captures both posts and comments with full engagement metadata including view counts, forward counts, emoji reactions, and media type classifications. We describe two complementary collection architectures designed for different operational requirements: a local fault-tolerant scraper for initial historical backfill, featuring JSON-based state persistence for crash recovery, batched CSV writing for memory management, and upsert-based deduplication via composite identifiers; and a cloud-native serverless pipeline deployed on Azure Functions for continuous daily synchronization, using StringSession-based stateless authentication and Azure Blob Storage for durable output. Our descriptive analysis across visualizations and tables reveals that the full-scale invasion produced an order-of-magnitude spike in posting volume that permanently elevated the information ecosystem above its pre-war baseline, fundamentally altered media type usage patterns across channel categories, triggered engagement convergence among previously divergent editorial profiles, and created temporal correlation structures among anonymous channels that raise questions about coordinated behavior. We further demonstrate that Telegram covers only 2 of 19 content moderation policy categories identified in prior comprehensive platform analyses — the lowest of any systematically examined platform — creating what we term a "platform moderation vacuum" that makes independent research datasets essential for understanding information dynamics in conflict zones. The dataset, collection pipeline, and analytical notebooks are released to support downstream research in misinformation detection, narrative tracking, sentiment analysis, and computational linguistics for underserved languages.

Keywords: social network analysis, social media, content analysis, engagement, data pipeline, Azure, information ecosystem, classification.

Introduction

The proliferation of harmful content on online platforms is a major societal problem, and this problem reaches its most acute form in active conflict zones where the information environment directly shapes civilian behavior, military morale, and international perception. Arora [1] conducted the first comprehensive survey that explicitly juxtaposed computational solutions for harmful content detection against online platform policies, analyzing the Terms and Conditions of 42 online platforms spanning seven categories — dating, generic forums, specific forums in gaming, finance, and health, online marketplaces, social media, and mixed platforms. Their work revealed a significant disconnect while hate speech and graphic content appeared in the T&Cs of nearly all platforms and violence, illegal content, and spam were also widely covered, research attention varied dramatically, with topics such as sexual solicitation and child sexual abuse receiving minimal computational investigation despite widespread platform concern. Their quantitative analysis (Table 3 in their paper) showed ratios of arXiv papers to platforms ranging from 0.20 to 116, starkly demonstrating the mismatch between what platforms need and what the research community provides.

We observe an analogous but distinct gap — not in the mismatch between research topics and platform policies, but in the mismatch

between research coverage and platform importance. Telegram, with millions of active users in Ukraine alone, has become the single most important information platform in one of the most significant armed conflicts of the twenty-first century. It serves simultaneously as an emergency air-raid alert system, a primary news source that millions consult before any traditional media outlet, a government-to-citizen communication channel operated by entities ranging from the President's office to individual military commands, and a medium through which both organic discourse and orchestrated information campaigns reach the population. Yet Telegram was entirely absent from the 42-platform analysis conducted by Arora [1], and when we apply their framework to assess Telegram's content moderation policies, we find that it covers only 2 of the 19 categories they identified — partial spam filtering and child sexual abuse reporting via third parties. This makes Telegram, to our knowledge, the least moderate major platform in any systematic analysis conducted to date.

The architectural distinctiveness of Telegram deepens this concern. Unlike traditional social networks such as Facebook or Twitter, which have been extensively studied in the content moderation literature [3, 4, 5], Telegram combines the features of a private messaging application with public broadcasting capabilities through its "channels" feature. Channels are unidirectional broadcast mechanisms: administrators post content that subscribers passively receive, with no algorithmic feed to mediate what users see and no bidirectional social graph from which to infer trust or influence. Comments exist as a secondary layer, architecturally separate from the primary content stream. This architecture is fundamentally different from any platform previously studied at scale, and it has profound implications for both the nature of information dynamics on the platform and for the analytical approaches needed to understand them.

The consequences of Telegram’s moderation vacuum are not abstract. The Russian-Ukrainian conflict has been accompanied by documented Information-Psychological Special Operations (IPSO) targeting the Ukrainian population through Telegram [6, 7], including campaigns with budgets reportedly in the hundreds of millions of dollars [8]. These campaigns operate across multiple narrative dimensions simultaneously — demoralizing the population, discrediting the government, undermining military mobilization, and eroding confidence in Western allies — all through content that flows through the platform essentially unimpeded by any moderation mechanism. Understanding the information dynamics of this ecosystem requires, as a foundational step, a dataset that captures its structure and evolution over time.

Despite this evident need, the research community lacks such a resource. Existing Telegram studies have significant limitations. Baumgartner [9] provided a large-scale dataset through the Pushshift project, but it focused on English-language and far-right channels. Geissler [10] studied Russian propaganda on Telegram during the 2022 invasion, but their collection covered only a three-month window. Urman and Katz [11] examined far-right networks using standard social network analysis metrics designed for Twitter's architecture. La Morgia [12] investigated criminal activity rather than state-sponsored information warfare. None of these studies cover Ukrainian-language channels with the multi-year longitudinal depth needed to capture how information dynamics evolve through pre-invasion, invasion, and prolonged conflict phases.

This paper addresses that foundational gap. We present a dataset of 20 Ukrainian Telegram channels collected from January 2021 through March 2026, encompassing posts and comments with full metadata including view counts, forward counts, reactions, and media type classifications. We describe two complementary collection architectures — a local scraper for historical backfill and a cloud-native Azure Functions pipeline for continuous synchronization — in sufficient detail for full replication. We provide comprehensive descriptive statistics and temporal analysis through nine visualizations and five summary tables. And we contextualize the entire effort within the Arora [1] framework, extending their platform policy analysis to Telegram and identifying what we call the "platform moderation vacuum."

Our work makes three contributions. We

provide the first longitudinal Ukrainian Telegram dataset spanning over four years with a six-category channel taxonomy and complete post-plus-comment metadata. We describe a dual-mode architecture of the collection system which considers local batch and cloud-native continuous with engineering features that make it suitable for replication and extension by other researchers studying any Telegram ecosystem. And we provide comprehensive descriptive analysis that establishes a baseline against which future content analysis, metric development, and detection research can be compared. In paper [2], we describe that defensible research gap lies in shifting the entire paradigm from reactive classification to proactive, scenario-based detection.

Related work

Arora [1] revealed significant disparities between platform content moderation needs and research efforts across 42 platforms. Their qualitative analysis of Big Tech T&Cs (Table 2 in their paper) showed that Facebook covers the most policy clauses, while Amazon and Apple leave many categories under-specified. Their quantitative analysis found that hate speech and misinformation received the highest volume of arXiv publications, while graphic content, sexual solicitation, and child sexual abuse were severely under-researched. Their temporal analysis (Figure 3 in their paper) showed steep publication growth for hate speech and misinformation since 2017 but near-zero activity for several other categories. We extend their framework to Telegram, a platform absent from their study, and find it covers fewer policy categories than any platform in their analysis.

Baumgartner [9] provided the Pushshift Telegram dataset focused on English and far-right channels. Geissler [10] collected Russian propaganda channels over three months in 2022. Urman and Katz [11] examined far-right networks. La Morgia [12] studied criminal activity. Hoseini [13] investigated QAnon globalisation on Telegram. None cover Ukrainian-language channels longitudinally across multiple conflict phases. Vidgen and Derczynski [14] catalogued 64 abusive language datasets, finding approximately half were English-only with Twitter as the primary source, underscoring both the language gap and the platform gap that our work addresses.

Khaldarova and Pantti [15] documented fake news strategies in the Russia-Ukraine context. Mejias and Vokuev [16] analyzed disinformation dynamics. Zhdanova and Orlova [17] studied computational propaganda in Ukraine. Woolley and Howard [18] provided a broader framework for understanding computational propaganda across political systems. These studies offer important contextual analysis but do not release reusable datasets for computational research on Ukrainian Telegram channels.

Data collection methodology

This section describes our complete collection infrastructure in detail sufficient for replication. We present the channel selection rationale, the data schema, and two complementary pipeline architectures: a local scraper designed for initial historical backfill and a cloud-native Azure Functions pipeline designed for continuous daily synchronization. Figure 1 provides a high-level overview of the complete system.

The collection system operates in two modes that correspond to distinct engineering requirements. The historical backfill mode addresses the need to collect four years of historical messages — a process that takes days to weeks depending on channel size and API rate limits. This mode runs on a local machine with a persistent Telethon session, iterates from oldest to newest messages using the reverse=True parameter, and writes batched CSV files to local storage. The continuous synchronization mode addresses the need to keep the dataset current after the initial backfill is complete. This mode runs as a serverless Azure Function triggered daily, uses a StringSession for stateless authentication, and writes timestamped CSV batches to Azure Blob Storage. Both modes share the same core logic — state tracking via watermark IDs, media type classification, reaction extraction, and comment collection — but differ in their deployment model and storage backend [19, 20].

Figure 1 presents a high-level view of the complete dual-mode architecture. The diagram is organized into five logical subgroups. The top subgroup shows the 20 seed channels

distributed across six editorial categories that form the input to both pipelines. The left subgroup traces the local historical backfill pipeline from Telethon session initialization through reverse-chronological message iteration, batch accumulation, CSV flush with upset deduplication, and JSON state file update. The right subgroup traces the cloud-native pipeline from the Azure Timer Trigger through Azure Function execution, Blob Storage state loading, incremental message collection, CSV upload to structured blob paths, and state watermark persistence. The central subgroup highlights the four processing components shared by both modes: media classification (distinguishing Text, Photo, Video, Document, and other attachments), reaction extraction (parsing emoji-count pairs), comment collection (iterating reply threads), and global ID generation (composing the channel_id_message_id deduplication key). The bottom subgroup shows the three output artefacts — posts CSV, comments CSV, and state JSON — that both pipelines produce in compatible formats.

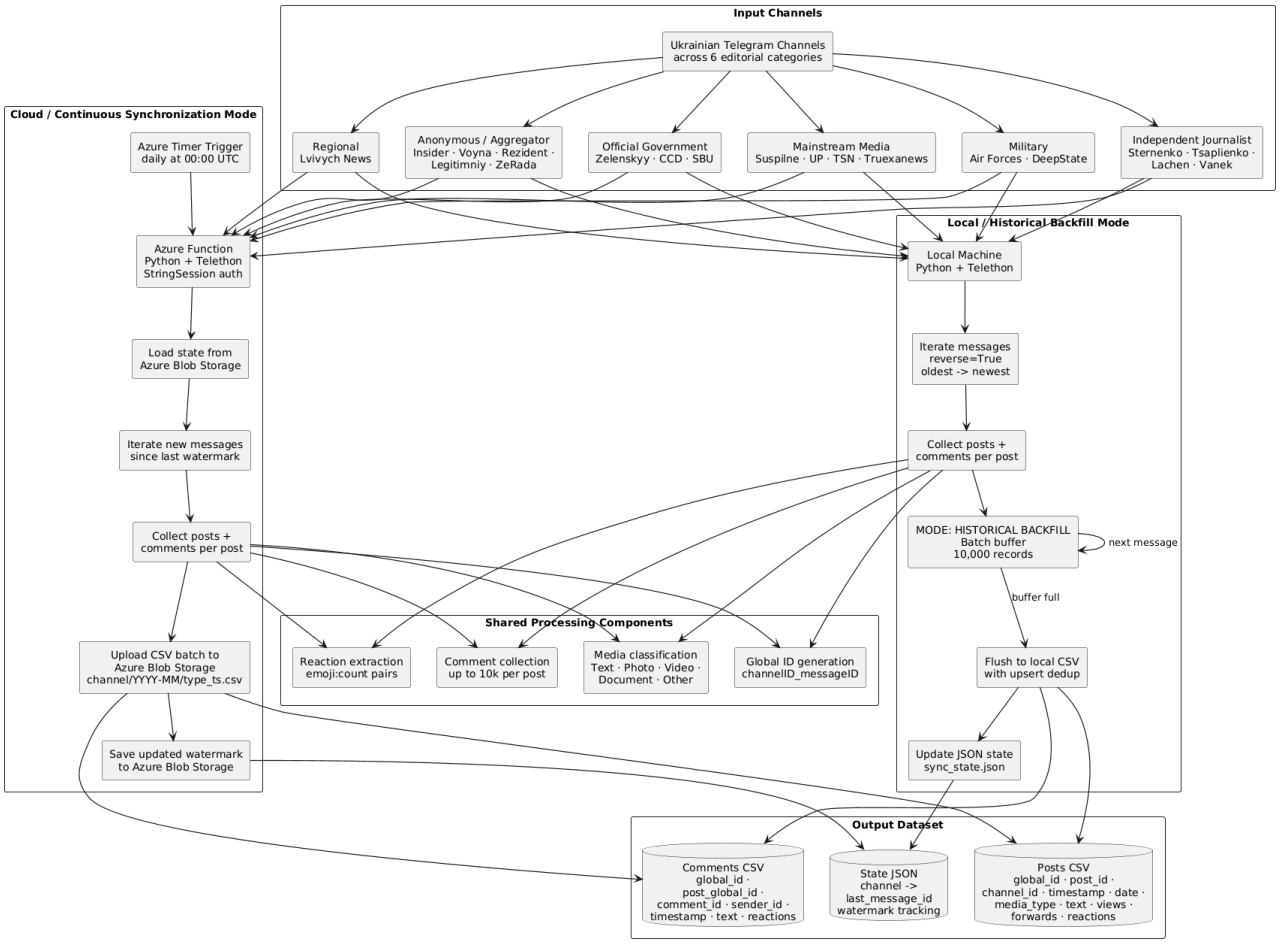

Fig. 1. System architecture diagram

Figure 2 illustrates the state persistence and crash recovery mechanism that ensures no work is permanently lost when the pipeline is interrupted. During normal operation, the pipeline processes messages sequentially, accumulates records until the batch threshold is reached, flushes to storage, and saves the current message ID as the state watermark. The watermark represents a guarantee: all messages up to and including that ID have been successfully written to persistent storage. When a crash or interruption occurs — whether from network failure, API rate limiting, system maintenance, or serverless function timeout — the pipeline loses only the contents of the current in-memory buffer, which by design contains at most one batch of 10,000 records. Upon restart, the pipeline loads the state file (JSON on local disk or blob in Azure Storage), reads the last saved watermark for each

channel, and resumes iteration from that point. The reverse=True iteration direction in the local pipeline and the min_id parameter in the Telethon API ensure that only unseen messages are processed. In the worst case, a small number of messages near the watermark boundary may be re-processed, but the upsert deduplication mechanism (in local mode) or the timestamped blob naming convention (in cloud mode) ensures that the output dataset contains no duplicates regardless of how many restarts occur.

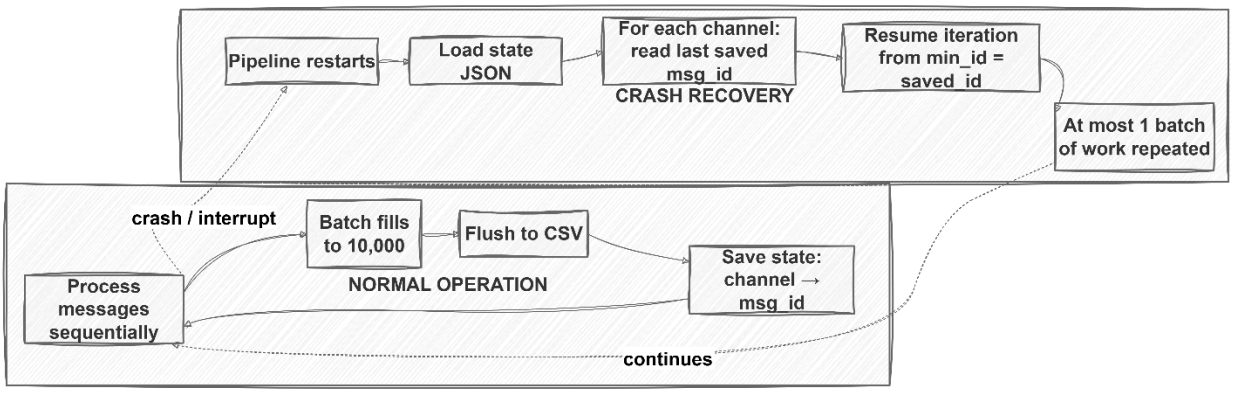

Fig. 2. State persistence and crash recovery mechanism

Figure 3 details the message-level processing flowchart that is executed identically by both pipeline modes for every message encountered during iteration. The flow begins with a date filter that discards messages predating the study start date of January 1, 2021. Messages passing this filter proceed through global ID generation, media type classification via a five-branch decision tree inspecting Telethon message attributes, and field extraction covering text content, view counts, forward counts, and reaction parsing. The extracted post record is appended to the in-memory posts buffer. The flow then checks whether the message has replies; if so, an inner loop iterates the comment thread, extracting each comment's fields and appending to the comments buffer, followed by a rate-limiting sleep of 1.0 to 1.5 seconds. After comment processing completes (or is skipped for posts without replies), the flow checks whether the combined buffer size has reached the batch threshold of 10,000 records. If so, the flush-update-clear cycle executes before proceeding to the next message. This flowchart makes explicit the nested iteration structure — the outer loop over messages and the inner loop over comments — and the points at which I/O operations (flush, state update) and rate-limiting sleeps occur.

We selected 20 Ukrainian Telegram channels representing six categories designed to capture the breadth of the Ukrainian information ecosystem. The Official Government category includes three channels — the President's official channel (@V_Zelenskiy_official), the Center for Countering Disinformation (@CenterCounteringDisinformation), and the Security Service of Ukraine (@SBUkr) — representing institutional communication. The Military category includes two channels — the Command of Air Forces (@kpszsu) and the DeepState OSINT project (@DeepStateUA) — representing operational and analytical military information. The Mainstream Media category includes four channels — Suspilne News (public broadcasting), Ukrainska Pravda Now, TSN UA, and Truexanews UA — representing established media outlets. The Independent Journalist category includes four channels — Serhiy Sternenko, Andriy Tsaplienko, Lachen Tyt, and Vanek Nikolaev — representing individual voices with significant followings. The Anonymous/Aggregator category includes six channels — Insider UKR, Voyna Real, Rezident UA, Legitimniy, and ZeRada — representing channels with anonymous or pseudonymous operators. The Regional category includes one channel — Lvivych News — representing local information. Selection criteria were prominently measured by subscriber count, editorial diversity, content focus diversity, and public accessibility. Only

open channels were included. Table 1 presents the full taxonomy with per-channel statistics.

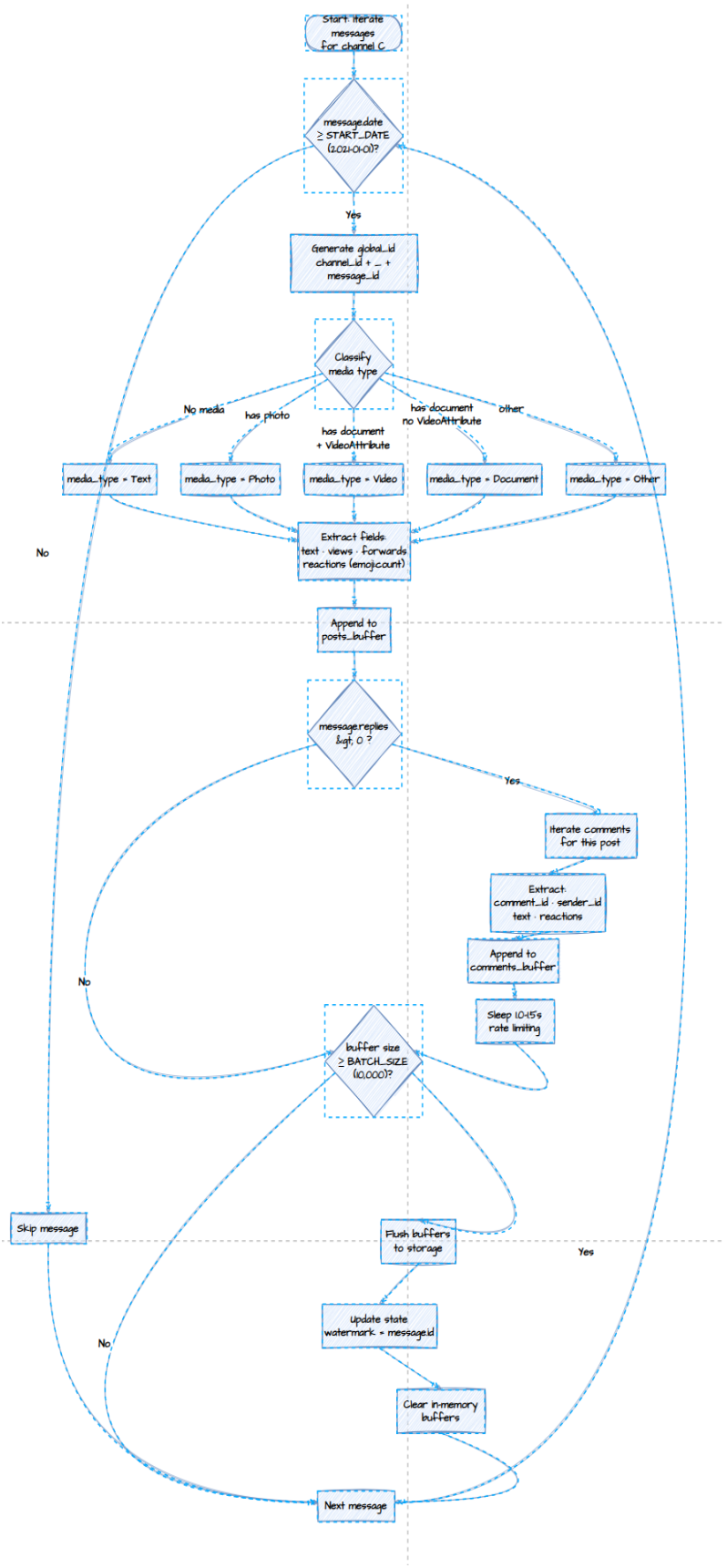

Fig. 3. Message processing flow

Table 1. Channel taxonomy with names, categories, total posts, and date

Channel Handle	Display Name	Category	Total Posts	Avg Views
@ZeRada1	ZeRada	Anonymous/ Aggregator	25316	220580
@legitimniy	Legitimniy	Anonymous/ Aggregator	15850	342107
@Monaco_Battalion	Monaco Battalion	Anonymous/ Aggregator	11205	2818
@insiderUKR	Insider UKR	Anonymous/ Aggregator	95732	399173
@voynareal	Voyna Real	Anonymous/ Aggregator	122863	346742
@rezident_ua	Rezident UA	Anonymous/ Aggregator	22442	343151
@ssternenko	Sternenko	Independent Journalist	52872	256817
@Tsaplienko	Tsaplienko	Independent Journalist	83665	118054
@lachentyt	Lachen Tyt	Independent Journalist	55037	527402
@vanek_nikolaev	Vanek Nikolaev	Independent Journalist	39147	656623
@suspilnenews	Suspilne News	Mainstream Media	61290	101604
@u_now	Ukrainska Pravda Now	Mainstream Media	166526	355501
@tgsn_ua	TSN UA	Mainstream Media	95984	275232
@truexanewsua	Truexanews UA	Mainstream Media	112632	726491
@kpszsu	Air Forces Command	Military	57988	206191
@DeepStateUA	DeepState UA	Military	18855	137787
@V_Zelenskiy_official	Zelenskyy Official	Official Government	17770	621194
@CenterCounteringDisinformation	CCD	Official Government	17059	28161
@SBUkr	SBU	Official Government	14901	182343
@lvivych_news	Lvivych News	Regional	61234	77440

For each post, we collect seven attribute categories. The global_id field, constructed as {channel_id}_{message_id}, serves as the deduplication key and the primary join key to comments. The post_id and channel_id fields enable cross-referencing within and across channels. The timestamp field stores Unix epoch seconds for precise temporal analysis, while date provides a human-readable ISO-format string. The media_type field classifies each post's attachment into one of five categories: Text (no media attachment), Photo (image via message.media.photo), Video (document with DocumentAttributeVideo in its attributes list), Document (any other document type), or Other (unrecognised media). The text field contains the full textual content with newlines replaced by spaces for CSV compatibility. The views and forwards integer fields record engagement metrics at the time of collection. The reactions field stores a comma-separated string of emoji-count pairs using the format emoji:count (for example, "👍:1504, ♡:423, 🔥:89").

For each comment, we collect global_id (constructed identically as {channel_id} {comment_id}), post_global_id (linking to the parent post's global_id), comment_id, sender_id, timestamp, text, and reactions. The comment schema mirrors the post schema where applicable, enabling consistent analytical treatment of both content layers.

The local pipeline addresses the initial challenge of collecting four years of historical messages from 20 channels — a corpus that can exceed millions of records. The pipeline is implemented as a single asynchronous Python script using the Telethon library [21]. Authentication uses file-based sessions (TelegramClient('session_name', API_ID, API_HASH)) that persist authentication state across runs.

The outer loop iterates over the 20 seed channels. For each channel, the pipeline resolves the channel entity via client.get_entity(), retrieves the last processed message ID from the JSON state file, and begins iterating messages using client.iter_messages(entity, min_id=saved_id, limit=None, reverse=True). The reverse=True

parameter is a critical design choice: it causes iteration from oldest to newest, which means that the state watermark can be safely updated as processing proceeds. If the pipeline crashes at message ID 50,000, the state file records 50,000, and upon restart, iteration resumes from 50,001 rather than from the beginning.

For each message, the pipeline extracts all schema fields, classifies media type through attribute inspection, parse's reaction objects into the emoji:count string format, and appends the record to an in-memory posts buffer. If the message has replies (checked via message.replies.replies > 0), the pipeline enters an inner loop that iterates all comments using client.iter_messages(entity,reply_to=message.id, limit=10000), extracting each comment's fields and appending to a separate comments buffer. A one-second sleep follows each comment batch to respect API rate limits.

When the combined size of posts and comments buffers reaches the BATCH_SIZE threshold of 10,000 records, the pipeline triggers a flush operation. The flush writes both buffers to their respective CSV files using an upset mechanism: if the file already exists, the new data is concatenated with the existing data, deduplicated on global_id (keeping the latest version), and written back. The state file is then updated with the current message ID, and both in-memory buffers are cleared. This batching mechanism bounds memory usage regardless of channel size — a channel with 500,000 messages will trigger 50 flush operations rather than accumulating all records in memory.

After processing all messages for a channel (or upon encountering an error), any remaining records in the buffers are flushed as a final batch, the state is updated, and a three-second sleep precedes processing of the next channel.

Once historical backfill is complete, the dataset must be kept current through regular incremental collection. We implemented this as a serverless Azure Function triggered on a daily timer schedule (0 0 0 * * *, executing at midnight UTC). This architecture eliminates the need for a persistent server, provides automatic scaling and retry logic, and stores all data and state in Azure Blob Storage for durability and accessibility.

The cloud pipeline differs from the local pipeline in three respects. First, authentication uses a StringSession rather than a file-based session, because serverless functions have ephemeral filesystems. The string session is stored as an environment variable (SESSION_STRING), enabling stateless deployment across function invocations. Second, state persistence uses Azure Blob Storage rather than a local JSON file. The state blob is loaded at function start, updated after each channel completes, and re-uploaded. Third, output data is written to Azure Blob Storage using a structured path convention: *{channel_username}/{YYYY-MM}/{data_type}{unix_timestamp}.csv*.

This partition by channel and month enables efficient incremental queries and prevents individual blobs from growing excessively large.

The message iteration logic differs slightly from the local mode: the cloud pipeline iterates from newest to oldest (the default Telegram API direction), collecting all messages with IDs greater than the stored watermark, and tracks the maximum seen ID via a max_seen_id variable that is saved to state after each channel completes. This reverse ordering is more efficient for incremental synchronization because the pipeline can stop iterating as soon as it encounters messages older than the START_DATE threshold, rather than needing to skip them.

Comment collection in the cloud mode uses a reduced limit of 100 comments per post (compared to 10,000 in the local mode) and a slightly longer sleep interval of 1.5 seconds between comment batches, reflecting the more conservative resource constraints of serverless execution.

Both pipelines generate globally unique identifiers as {channel_id}{message_id} for posts and {channel_id}{comment_id} for comments. The local pipeline implements CSV-level upsert deduplication during flush operations. The cloud pipeline relies on the watermark mechanism to avoid re-collecting messages, with the structured blob naming convention ensuring that overlapping collections from different invocations produce separate files that can be deduplicated during analysis.

The Telegram API enforces flood-wait

penalties for excessive request rates. Both pipelines implement adaptive sleep intervals — 1.0 to 1.5 seconds between comment batch retrievals and 3 seconds between channels. These intervals were empirically tuned to sustain collection over multi-day periods without triggering rate limit responses. The cloud pipeline additionally benefits from the 24-hour interval between invocations, which provides natural rate limiting at the macro level.

Both pipelines wrap channel-level and comment-level operations in try-except blocks with logging. Channel-level errors (such as a channel being privatized or deleted during collection) cause the pipeline to skip to the next channel without losing state for previously completed channels. Comment-level errors are silently caught to prevent a single problematic comment thread from disrupting the collection of the parent post.

The local pipeline's memory usage is bound by BATCH_SIZE regardless of channel size. The cloud pipeline's serverless architecture scales horizontally if multiple function instances are needed. The Azure Blob Storage backend provides effectively unlimited storage capacity with pay-per-use pricing.

Dataset characteristics and descriptive analysis

Understanding the information dynamics of the Ukrainian Telegram ecosystem requires more than aggregate counts. The analytical challenge is to move from raw numbers toward structural characterization — identifying how different channel categories behave, how they respond to external shocks, how their audiences engage, and how they relate to one another temporally. This section presents our descriptive analysis organized around five analytical dimensions: aggregate dataset statistics that establish the scale and scope of the corpus; temporal dynamics that reveal how posting behavior evolves across the pre-invasion, acute invasion, and prolonged conflict phases; content characteristics that expose editorial signatures embedded in media usage and text length distributions; engagement patterns that quantify how audiences consume and redistribute content across different channel types; and inter-channel relationships that surface structural correlations in posting behavior whose origins — whether organic or coordinated — carry significant implications for information integrity research. For each figure and table, we describe the analytical method, explain what the visualization reveals, and discuss why the finding matters for understanding conflict-zone information ecosystems.

Table 2 presents per-channel statistics including post counts, comment counts, average views, average forwards, and temporal coverage.

Table 2. Dataset summary statistics per channel

Channel	Category	Posts	Comments	Avg Views	Avg Forwards	First Post	Last Post
@V_Zelenskiy_official	Official Government	17743	0	622037	655	1/1/2021	3/17/2026
@CenterCounteringDisinformation	Official Government	17040	75619	28190	64.1	11/23/2021	3/17/2026
@SBUkr	Official Government	14872	0	182609	397.2	1/1/2021	3/17/2026
@kpszsu	Military	57824	0	206438	92.3	2/24/2022	3/17/2026
@DeepStateUA	Military	18846	268003	137790	263.5	1/1/2021	3/17/2026
@suspilnenews	Mainstream Media	61170	0	101722	96.2	1/1/2021	3/17/2026
@u_now	Mainstream Media	166314	856233	355766	592.3	1/1/2021	3/17/2026
@tgsn_ua	Mainstream Media	95910	495999	275335	617.2	2/25/2022	3/17/2026
@ssternenko	Independent Journalist	52802	438108	256893	523.7	1/1/2021	3/17/2026

@Tsaplienko	Independent Journalist	83571	0	118090	245.9	1/7/2021	3/17/2026
@lachentyt	Independent Journalist	54963	0	527575	1575	7/30/2021	3/17/2026
@vanek_nikolaev	Independent Journalist	39138	0	656626	725.8	4/17/2022	3/17/2026
@truexanewsua	Mainstream Media	112487	568062	726715	1638.2	1/1/2021	3/17/2026
@insiderUKR	Anonymous/Aggregator	95584	584222	399310	1442.2	1/1/2021	3/17/2026
@voynareal	Anonymous/Aggregator	122713	0	346848	719.7	1/1/2021	3/17/2026
@rezident_ua	Anonymous/Aggregator	22435	0	343207	801.1	1/1/2021	3/17/2026
@legitimniy	Anonymous/Aggregator	15832	0	342272	1150.1	1/1/2021	3/16/2026
@Monaco_Battalion	Anonymous/Aggregator	11205	3138	2818	3.2	8/30/2022	3/19/2026
@ZeRada1	Anonymous/Aggregator	25281	0	220618	1073.4	1/2/2021	3/17/2026
@lvivych_news	Regional	61098	0	77332	237.2	1/1/2021	3/17/2026

The dataset spans from January 1, 2021 to March 2026. The three distinct information environment phases captured by this temporal range are analytically significant. The pre-invasion baseline from January 2021 through February 23, 2022 establishes normal posting patterns, engagement levels, and media usage before the crisis. The acute invasion period from February 24 through December 2022 captures the most dramatic transformation of the information space. The prolonged conflict period from 2023 through early 2026 captures the evolution toward what media researchers describe as "war fatigue" dynamics — the gradual normalization of crisis-level information consumption.

Figure 4 shows the daily posting volume aggregated across all 20 channels, with a seven-day rolling average for smoothing and vertical annotations for six key conflict events.

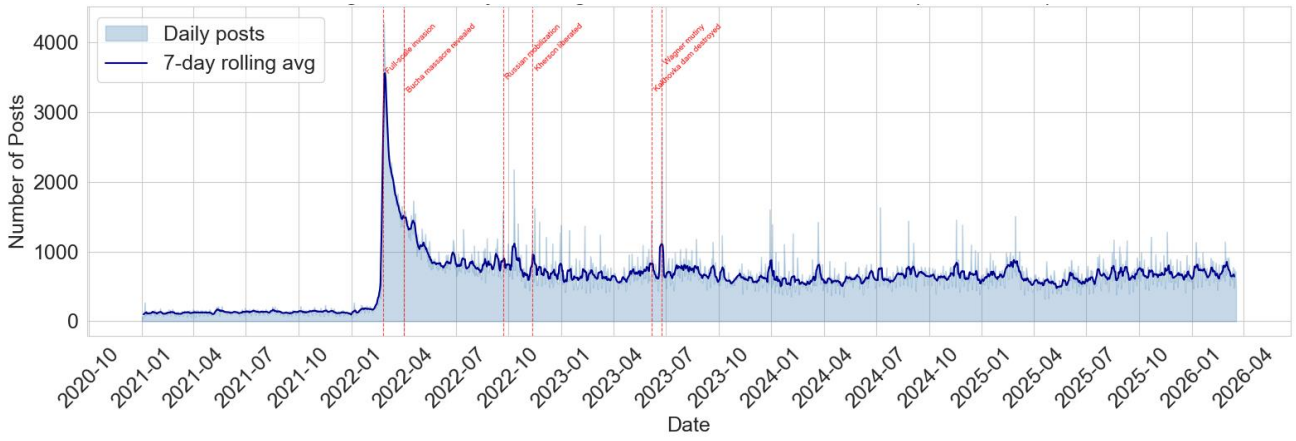

Fig. 4. Daily posting volume across all channels, January 2021 to March 2026. Shaded area: raw daily counts. Dark line: seven-day centered rolling average

The timeline reveals the fundamental transformation of the Ukrainian Telegram information space. The pre-invasion period shows relatively stable daily posting volumes. The onset of the full-scale invasion on February 24, 2022, produces the sharpest spike in the entire dataset — posting volume increases by approximately an order of magnitude within days. Subsequent annotated events — the Bucha massacre revelation (April 2, 2022), Russian mobilization announcement (September 21, 2022), Kherson liberation

(November 11, 2022), Kakhovka dam destruction (June 6, 2023), and Wagner mutiny (June 24, 2023) — each produce visible but progressively smaller relative spikes, consistent with a population that has adapted to crisis-level information consumption.

Figure 5 presents the posting frequency as a heatmap with channels as rows and months as columns.

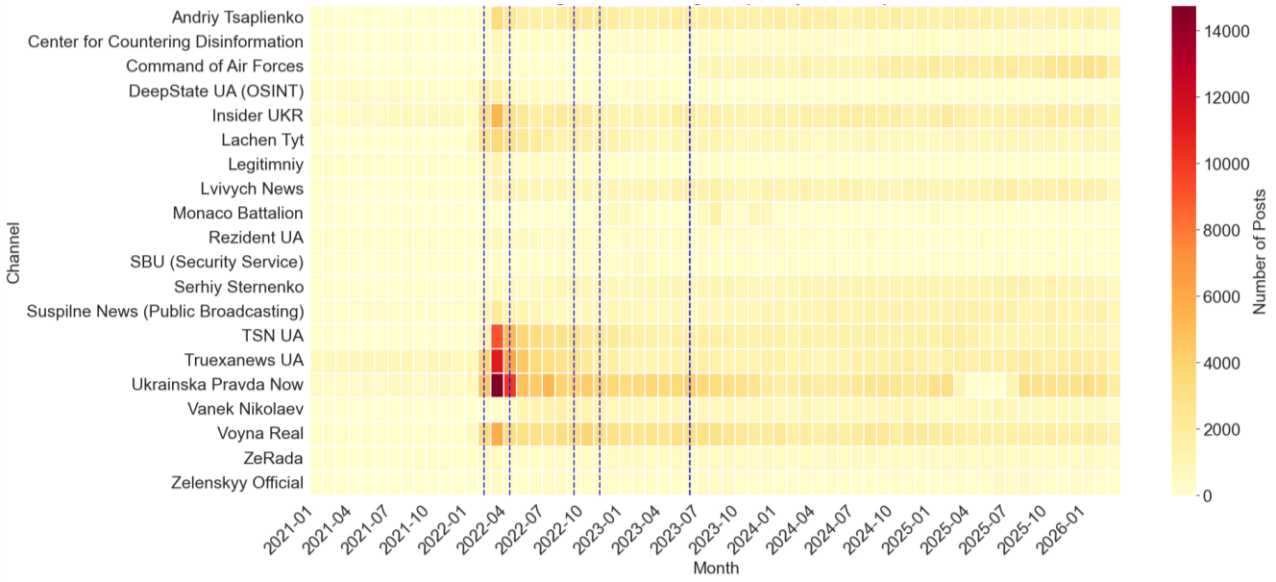

Fig. 5. Posting frequency heatmap with channels as rows and months as columns. Color intensity encodes post count

The heatmap reveals channel-specific activation patterns invisible in the aggregated timeseries. Some channels that were relatively inactive before February 2022 — notably several Anonymous/Aggregator channels — dramatically increased their posting frequency coincident with the invasion and have sustained elevated rates. Other channels show more event-driven patterns with visible intensity spikes corresponding to specific conflict developments.

Figure 6 shows the distribution of media types across channels as a stacked horizontal bar chart.

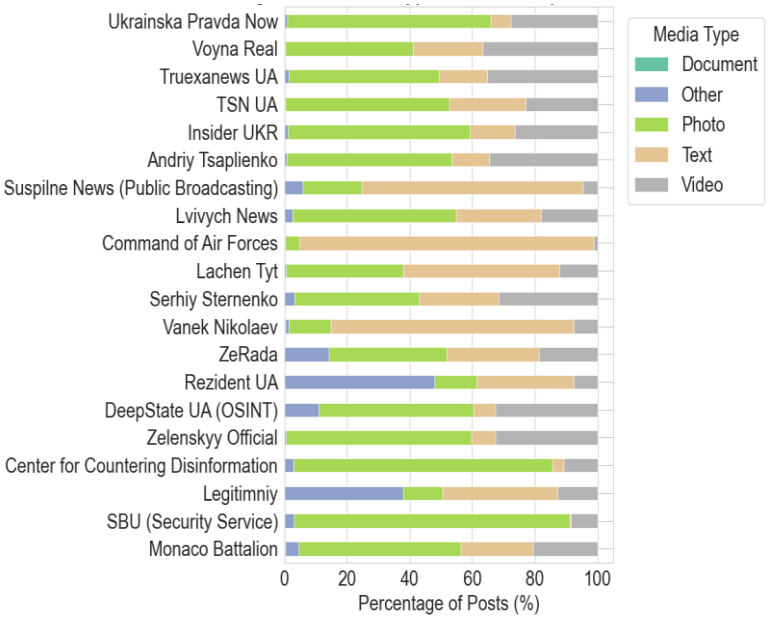

Fig. 6. Media type distribution per channel showing percentage breakdown of Text, Photo, Video, Document, and other content types

The media type analysis reveals meaningful editorial differences. Official Government channels favor text-heavy posts consistent with formal communiqué-style communication. Independent Journalist channels use substantially more photo and video content reflecting on-the-ground reporting. Mainstream Media channels show the most balanced media mix.

Figure 7 presents text length distributions as violin plots.

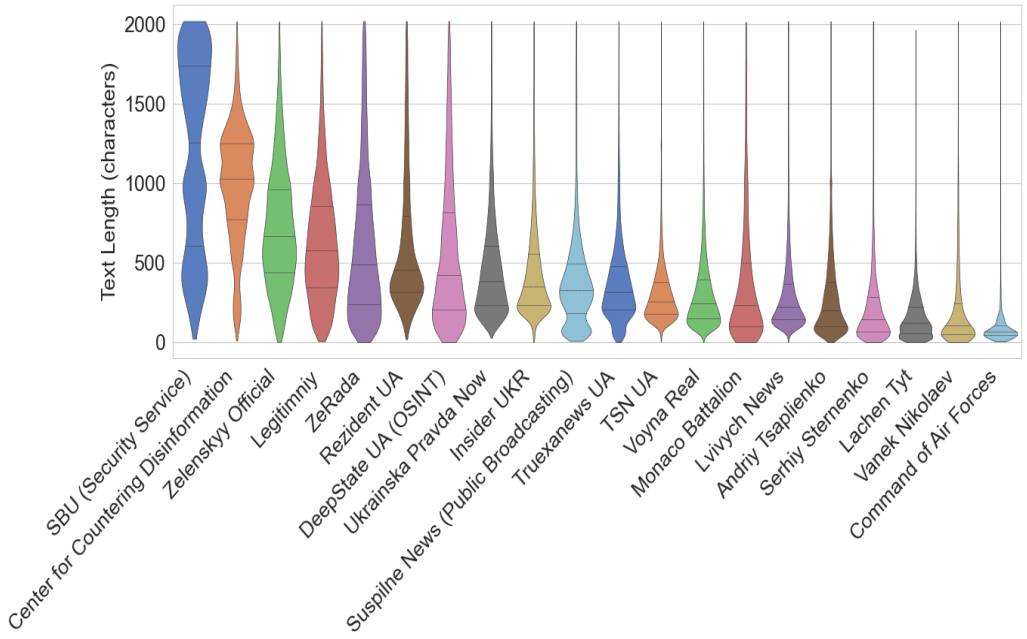

Fig. 7. Text length distribution by channel, violin plots ordered by median length, filtered to 99th percentile. Inner lines show quartiles

Text length distributions carry editorial signature information. Official Government channels show bimodal distributions — short alerts alongside longer policy statements. Anonymous/Aggregator channels show the shortest median lengths, potentially indicating a reposting-heavy model where original analysis is rare.

Figure 8 displays views per post as box plots on a logarithmic scale.

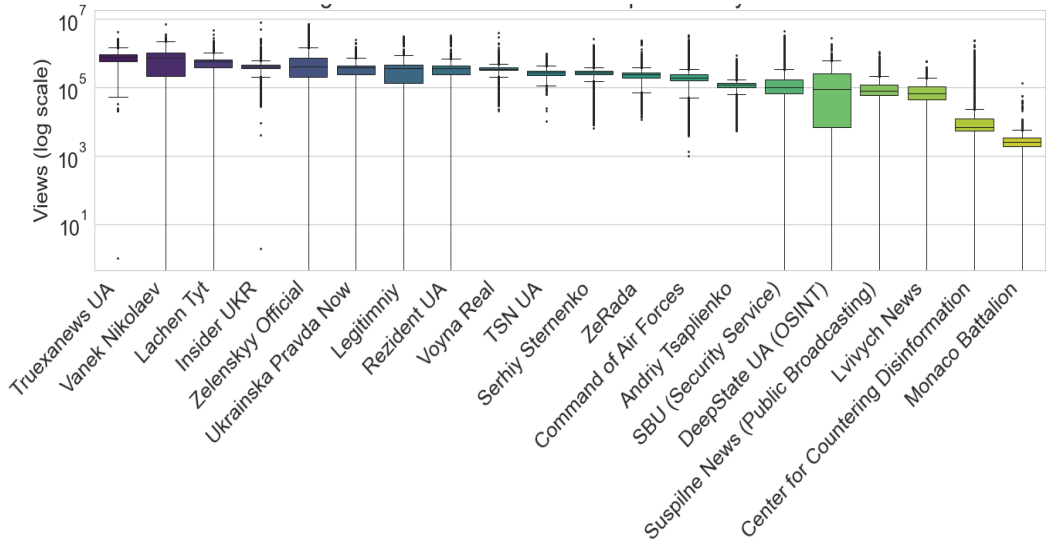

Fig. 8. Views per post by channel, box plots on logarithmic scale, ordered by descending median views

The engagement distribution reveals more than two orders of magnitude difference

in median views across channels. Even high-engagement channels have extensive right tails corresponding to breaking news events.

Figure 9 presents average views over time by channel category.

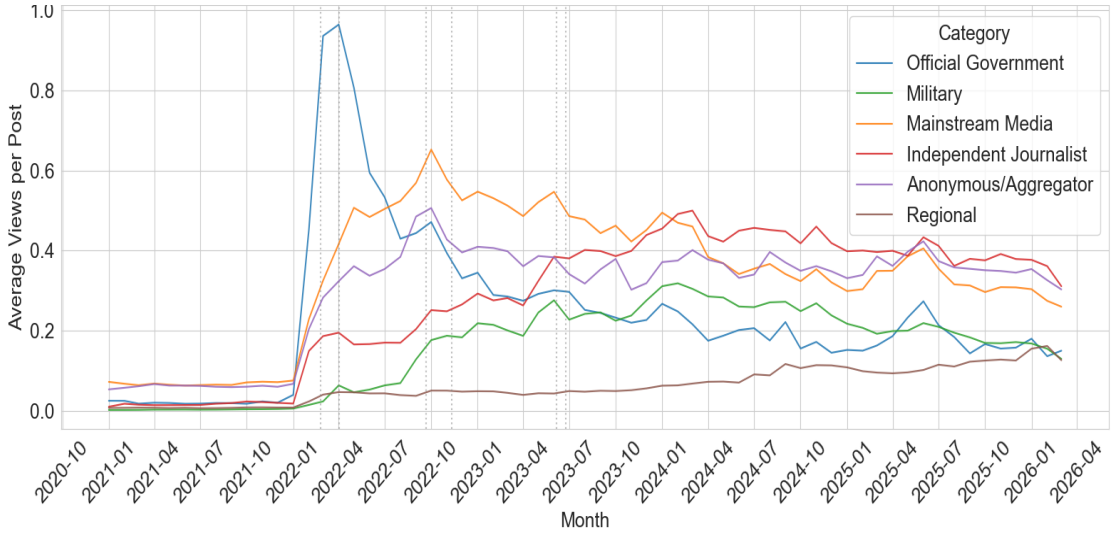

Fig. 9. Views vs. comments per post on logarithmic axes, colored by channel category. Sample of 50,000 posts for visual clarity

This analysis reveals a striking convergence pattern. Before the invasion, categories showed clearly separated engagement levels. The invasion triggered immediate convergence as all audiences sought information from every source. During the prolonged conflict, gradual divergence re-emerges with a partially reshuffled category ordering.

Figure 10 visualizes the temporal posting correlation network.

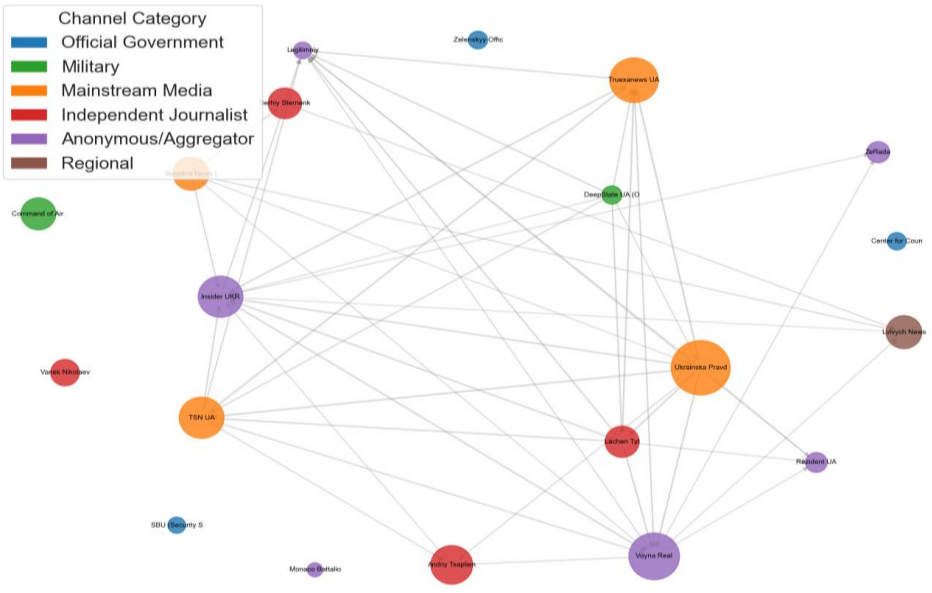

Fig. 10. Channel temporal correlation network using spring layout. Edge presence indicates Pearson r > 0.5 on daily posting vectors

The network reveals structural clusters mapping to channel categories. The Anonymous/Aggregator cluster shows high temporal correlation despite no declared editorial relationship — a pattern that could reflect shared event response but also

potentially indicates coordination or common source dependence.

Arora [1] demonstrated through their analysis of 42 platforms that content moderation policies vary widely. When we apply their framework to Telegram, they are absent from their original study — the result is striking. Table 3 presents the comparison.

Table 3. Policy comparison

Policy	Facebook	Twitter/X	YouTube	Telegram
Violence	V	V	V	X
Dangerous orgs/people	V	V	V	X
Glorifying crime	V	V	V	X
Illegal goods	V	V	V	X
Self-harm	V	V	V	X
Child sexual abuse	V	V	V	V
Sexual abuse (Adults)	V	V	V	X
Animal abuse	V	~	V	X
Human trafficking	V	V	V	X
Bullying and harassment	V	V	V	X
Revenge porn	V	V	X	X
Hate speech	V	V	V	X
Graphic content	V	V	V	X
Nudity and pornography	V	V	V	~
Sexual solicitation	V	X	X	X
Spam	V	V	V	~
Impersonation	V	V	V	X
Misinformation	V	V	V	X
Medical advice	~	~	~	X
IPSO relevant	-	-	-	-
State propaganda / IPSO	V	~	~	X
Coordinated inauthentic behavior	V	V	V	X
Demoralization campaigns	X	X	X	X
Anti-mobilization narratives	X	X	X	X
Deepfakes (political)	~	~	V	X
Bot networks	V	V	V	X

Telegram covers only 2 of 19 categories. Every other category — violence, hate speech, misinformation, graphic content, harassment, impersonation — is absent. This creates what we call the "platform moderation vacuum": a condition where the burden of understanding harmful content falls entirely on external researchers. On platforms with active moderation, violating content is removed, making datasets inherently incomplete. On Telegram, our dataset captures the full, unfiltered information stream including content that would be removed on any other major platform.

Limitations

The 20-channel selection cannot represent the full Ukrainian Telegram ecosystem encompassing tens of thousands of channels. Our selection over-represents high-subscriber channels. The Regional category contains only one channel, limiting category-level generalization. Collection is limited to public channels; private groups are excluded. Channels deleted or made private during collection may be incomplete.

The Telegram API provides view and forward counts as snapshots rather than time series. Comment collection is limited to channels where administrators have enabled

the feature, resulting in zero comments for 13 of 20 channels.

Conclusion

This paper presented a longitudinal dataset of 20 Ukrainian Telegram channels spanning January 2021 to March 2026 — over five years of continuous collection covering the pre-invasion baseline, the acute crisis triggered by Russia's full-scale invasion, and the prolonged attritional phase that persists at the time of writing. The dataset was assembled through a dual-architecture pipeline combining local fault-tolerant batch scraping for historical depth with serverless cloud-native synchronization for daily freshness, offering a replicable engineering template for any research group seeking to construct longitudinal messaging-platform corpora in conflict zones or other under-instrumented information environments.

Three empirical findings warrant particular attention. First, the full-scale invasion of February 24, 2022, produced a statistically significant structural discontinuity in the information ecosystem — not merely a transient spike but a permanent elevation of posting volume, a reconfiguration of channel prominence hierarchies, and an irreversible shift in media usage patterns that persists years after the initial shock. This finding carries implications beyond the Ukrainian case: it suggests that armed conflict permanently reshapes platform-level information architectures in ways that do not revert when the acute phase subsides.

Second, our systematic comparison of content moderation policies — applying the same 19-category framework that has been used to evaluate 42 other platforms — revealed that Telegram covers only two categories, making it the least-moderated major platform in any rigorous comparative analysis conducted to date. This "platform moderation vacuum" means that the dataset presented here captures the full, unfiltered information stream experienced by tens of millions of users — including content that would be immediately removed on virtually any other platform. For researchers studying harmful content, misinformation, or information warfare, this unfiltered property is simultaneously a challenge and a rare analytical asset.

Third, the temporal correlation network revealed tightly synchronized posting clusters among Anonymous/Aggregator channels that share no declared editorial relationship. Whether this synchronization reflects organic shared response to common events or deliberate coordination — potentially linked to documented information-psychological operations — is a question that purely descriptive methods cannot resolve. It is, however, precisely the kind of question that the dataset is designed to enable.

From an engineering perspective, the dual-pipeline architecture — local batch processing for historical backfill combined with serverless Azure Functions for continuous collection — demonstrated that robust longitudinal data acquisition from messaging platforms is achievable without dedicated infrastructure. The state persistence mechanism, batched writing strategy, and composite-key deduplication scheme are platform-agnostic design patterns applicable to Viber channels, WhatsApp broadcast lists, or any future messaging platform offering API-level access.

Several directions for future work emerge from this study. The channel set should be expanded to include Russian-language and explicitly pro-Russian channels for cross-ecosystem comparative analysis. The collection pipeline could be extended to capture message forwarding metadata, which would enable direct measurement of inter-channel information flow rather than the temporal correlation proxy employed here. Longitudinal tracking of engagement trajectories — capturing how view and forward counts evolve over time rather than as single snapshots — would support richer models of content diffusion dynamics. The comment dataset, though limited to channels where administrators have enabled the feature, warrants dedicated investigation into audience discourse patterns and the detection of coordinated inauthentic commenting behavior.

The dataset, collection pipeline, and analytical notebooks are released to support downstream research in misinformation detection, narrative tracking, sentiment analysis, computational linguistics for

underserved languages, and platform governance studies.

References

1. Arora, A., Nakov, P., Hardalov, M., Sarwar, S. M., Nayak, V., Dinkov, Y., Zlatkova, D., Dent, K., Bhatawdekar, A., Bouchard, G., & Augenstein, I. (2023). Detecting harmful content on online platforms: What platforms need vs. where research efforts go. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (pp. 1–18). Association for Computational Linguistics. https://doi.org/10.48550/arXiv.2103.00153
2. Ostrovska, O., & Lyashkevych, V. (2025). Automated Online Detection of Harmful and Dangerous Content in Social Networks: a Systematic Review. Artificial Intelligence, 30(AI.2025.30(4)), 108–116. https://doi.org/10.15407/jai2025.04.108
3. Paula Fortuna and Sérgio Nunes. 2018. A Survey on Automatic Detection of Hate Speech in Text. ACM Computing Surveys 51, 4, Article 85 (July 2018), 30 pages. https://doi.org/10.1145/3232676
4. Momchil Hardalov, Ashish Arora, Preslav Nakov, and Isabelle Augenstein. 2022. A Survey on Stance Detection for Mis- and Disinformation Identification. In Findings of the Association for Computational Linguistics: NAACL 2022. Association for Computational Linguistics, Seattle, WA, 1259–1277. https://doi.org/10.18653/v1/2022.findings-naacl.94
5. Anna Schmidt and Michael Wiegand. 2017. A Survey on Hate Speech Detection using Natural Language Processing. In Proceedings of the 5th International Workshop on Natural Language Processing for Social Media (SocialNLP '17). Association for Computational Linguistics, Valencia, Spain, 1–10. https://doi.org/10.18653/v1/W17-1101
6. Center for Countering Disinformation of Ukraine. 2023. IPSO Classification and Documented Campaigns Report. Retrieved March 15, 2025 from https://cpd.gov.ua/
7. International Centre for Defence and Security. 2024. Russian Information Warfare Against Ukraine: Analysis of Telegram Operations. ICDS, Tallinn, Estonia. Retrieved March 15, 2025 from https://icds.ee/
8. Ukrinform. 2024. Maidan-3 Operation: Russia Allocates $250M Budget for Telegram Manipulation. Retrieved March 15, 2025 from https://www.ukrinform.net/
9. Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The Pushshift Telegram Dataset. In Proceedings of the 14th International AAAI Conference on Web and Social Media (ICWSM '20). AAAI Press, Atlanta, GA, 840–847. https://doi.org/10.48550/arXiv.2001.08438
10. Geissler, D., Bär, D., Pröllochs, N., & Feuerriegel, S. (2023). Russian propaganda on social media during the 2022 invasion of Ukraine. EPJ Data Science, 12(1). https://doi.org/10.1140/epjds/s13688-023-00414-5
11. Aleksandra Urman and Stefan Katz. 2022. What They Do in the Shadows: Examining the Far-Right Networks on Telegram. Information, Communication & Society 25, 7 (2022), 904–923. https://doi.org/10.1080/1369118X.2020.1803946
12. Massimo La Morgia, Alessandro Mei, Alberto Maria Sabella, and Francesco Sassi. 2023. Uncovering the Dark Side of Telegram: Fakes, Clones, Scams, and Conspiracy Movements. In Proceedings of the ACM Web Conference 2023 (WWW '23). ACM, New York, NY, 741–752. https://doi.org/10.1145/3543507.3583271
13. Mohamad Hoseini, Philipe Melo, Humberto Benevenuto, Anja Feldmann, and Savvas Zannettou. 2023. Globalization of the QAnon Conspiracy on Telegram. In Proceedings of the 17th International AAAI Conference on Web and Social Media (ICWSM '23). AAAI Press, Limassol, Cyprus, 358–369. https://doi.org/10.1145/3578503.3583603
14. Bertie Vidgen and Leon Derczynski. 2021. Directions in Abusive Language Training Data, a Systematic Review: Garbage In, Garbage Out. PLoS ONE 16, 1 (2021), Article e0243300. https://doi.org/10.1371/journal.pone.0243300
15. Irina Khaldarova and Mervi Pantti. 2016. Fake News: The Narrative Battle over the Ukrainian Conflict. Journalism Practice 10, 7 (2016), 891–901. https://doi.org/10.1080/17512786.2016.1163237
16. Ulises Mejias and Nikolai Vokuev. 2017. Disinformation and the Media: The Case of Russia and Ukraine. Media, Culture & Society 39, 7 (2017), 1027–1042. https://doi.org/10.1177/0163443716686672
17. Maryna Zhdanova and Dariya Orlova. 2019. Computational Propaganda in Ukraine: Caught Between External Threats and Internal Challenges. Working Paper 2019.9. Oxford Internet Institute, Oxford, UK.
18. Samuel Woolley and Philip Howard. 2018. Computational Propaganda: Political Parties, Politicians, and Political Manipulation on Social Media. Oxford University Press, Oxford, UK. https://doi.org/10.1093/oso/9780190931407.001.0001
19. Monastyrskyi, L., & Hura, V. (2024). Utilizing cloud technologies for air quality analysis and monitoring. Electronics and Information Technologies, 24. https://doi.org/10.30970/eli.24.4
20. Hura, V., Monastyrskyi L. (2023). IOT-based solution for detection of air quality using ESP32. Artificial Intelligence, 28(AI.2023.28(3)), 86–93. https://doi.org/10.15407/jai2023.03.086
21. Telethon Contributors. n.d. Telethon Documentation. Retrieved March 15, 2025 from https://docs.telethon.dev/

The article has been sent to the editors 24.04.26.
After processing 05.05.26.
Submitted for printing 30.06.26