Scrolling is not free: The hidden cost of Shorts

Jamil Abou Ltaif, Chadi Barakat, Frédéric Giroire, Joanna Moulierac, Thierry Turletti
Université Côte d'Azur, Inria, CNRS - France

***Abstract*—Short videos have become one of the most popular forms of content consumption on mobile devices. Unlike traditional video streaming, users often watch many short videos within a single session and frequently swipe away videos after only a brief viewing period. This interaction pattern can lead to significant data waste, where video content is downloaded but never viewed. In this paper, we systematically investigate data waste in short video streaming and propose an experimental methodology to quantify its magnitude and underlying causes. Using YouTube Shorts as a case study, we design multiple controlled viewing scenarios to measure the fraction of downloaded video data that is not watched. We further analyze how video properties, content categories, and the platform's buffering and prefetching strategies contribute to this waste. Our results reveal substantial inefficiencies in short video delivery, highlighting opportunities for improving data efficiency in short-form video streaming systems.**

***Index Terms*—Short Video, Data Waste, Swiping Behavior.**

I. Introduction

Video streaming accounts for more than 70% of global Internet traffic [1], [2], highlighting its dominant role in today's digital ecosystem. In recent years, short vertical videos have rapidly transformed mobile content consumption. Platforms such as YouTube Shorts, TikTok, and Instagram Reels are leading this shift [3]. More than one billion users watch short videos every month, and the global market for short video platforms is expected to double between 2022 and 2030 [4].

YouTube is currently the largest video-sharing platform worldwide. As of 2025, YouTube Shorts alone reaches over 70 billion daily views and engages more than 2 billion monthly active users (MAUs) [5], [6]. At the Cannes Lions 2025 event, YouTube CEO announced a new milestone: "YouTube Shorts are now averaging over 200 billion daily views" [7]. The rapid growth of short videos is expected to make them account for nearly 80% of all mobile video traffic by 2025 [2]. These trends indicate increasing pressure on mobile networks and cloud infrastructures, raising concerns about network capacity, data efficiency, and device energy consumption.

While platforms such as TikTok and Instagram Reels pioneered the short-form video format and continue to strongly influence user behavior, YouTube Shorts remains comparatively underexplored in empirical research. Given its growing prominence and the limited attention it has received, this study aims to fill this gap by analyzing the streaming behavior and performance of YouTube Shorts. In addition, YouTube Shorts is integrated into YouTube's infrastructure and ecosystem, offering a unique opportunity to study short-form delivery on a platform built for long-form streaming.

ISBN 978-3-903176-82-9 © 2026 IFIP

Short videos, typically defined as videos shorter than 60 s, are mainly consumed through smartphone applications. They are displayed in a full-screen, vertical format optimized for swipe-based navigation. Users move through a continuous playlist by swiping vertically to access the next or previous video. The order of videos is usually determined by recommendation algorithms. Each video is stored on a streaming server and divided into multiple segments, which may be encoded at different quality levels to adapt to changing network conditions [8]. The user agent, typically a mobile application, selects and downloads these segments using standard streaming protocols such as DASH and HLS.

Unlike traditional long-form videos, short-form videos are designed for fast and lightweight interaction. Users often watch only a few seconds of a video before swiping to the next one, driven by auto-play features and recommendation systems. To reduce playback delay and ensure a smooth user experience, platforms proactively prefetch video segments before users fully commit to watching. While this improves responsiveness, it frequently leads to unnecessary data downloads, as many prefetched segments are never viewed. As a result, a large amount of video data is transferred to user devices but remains unwatched, leading to significant data and energy waste [9], [10], [11], [12], [13], [14], [15].

This problem, despite its scale, has been overlooked by the networking and systems research community. Most existing work on adaptive video streaming assumes longer viewing sessions and continuous playback behavior [16], which do not reflect the bursty and unpredictable nature of short-form consumption. In addition, few studies analyze the protocol-level effects of frequent video swiping, where reducing unnecessary data transfers is critical for scalable and efficient delivery. Existing approaches primarily focus on ensuring smooth playback through aggressive prefetching, rather than directly measuring or minimizing data waste. This creates a growing mismatch between how streaming systems are designed and how users actually interact with mobile-first video platforms. Although recent works such as Dashlet [12], PDAS [17], and Sprinkle [13] begin to account for swiping behavior and adaptive preloading, they often rely on isolated optimizations or simulated traces and do not provide a comprehensive evaluation of real-world data waste.

In this paper, we address this gap by presenting the first in-depth analysis of short-form video streaming at the protocol level and its impact on data usage efficiency. We quantify the amount of unnecessary data transferred due to frequent swiping behavior and examine how existing adaptive streaming mechanisms perform under highly bursty and short viewing sessions. Beyond measurement, we also explore practical strategies to reduce this waste by redesigning how video segments are prefetched, buffered, and delivered. Using YouTube Shorts as a real-world case study, we evaluate multiple lightweight policies that adapt downloading behavior to actual user watch time and demonstrate their potential to significantly reduce unused data.

Our contributions can be summarized as follows:

- We collect and analyze a large-scale dataset of over 100 000 YouTube Shorts spanning 15 content categories, enabling an empirical investigation of short-form video streaming behavior at scale. We make this dataset publicly available to support reproducibility and future research [1].
- We perform a measurement-driven study of playback behavior under different watch durations, ranging from rapid swipes to full video views, and quantify the resulting data waste caused by unused prefetched and buffered segments. All experiments are conducted in a fully controlled network environment, allowing us to precisely isolate platform-side streaming behavior from external network variability.
- We analyze buffer dynamics and identify a clear mismatch between existing prefetching and buffering strategies and actual user viewing behavior. Our results show that aggressive preloading policies often lead to temporal inflation of buffered content and substantial data inefficiency in short-form viewing scenarios.
- We investigate how video characteristics, such as bitrate variability and content category, affect data waste, and demonstrate that a significant portion of the waste originates from the delivery of early video segments that are unlikely to be fully watched.
- Building on these insights, we propose and evaluate a set of lightweight, swipe-aware delivery policies that adapt prefetching, first-segment downloading, and buffering decisions to observed user watch time. Our results show that these policies can substantially reduce unnecessary data transfer without compromising playback responsiveness.

Our findings highlight the need for a shift in how mobile video streaming systems are designed. Beyond minimizing startup delay and ensuring smooth playback, data efficiency and sustainability must become first-class design goals. By aligning data delivery more closely with user behavior, especially in short-form, swipe-heavy scenarios, platforms can significantly reduce wasted data and associated energy consumption. These insights are critical for building more efficient and user-aware streaming systems for future mobile networks.

The rest of the paper is organized as follows. Section II reviews related work and summarizes the state of the art in short-form and adaptive video streaming. Section III presents our measurement methodology, including the testbed design, data collection process, and metrics used to quantify data waste and playback efficiency. Section IV reports our measurement results and evaluates the effectiveness of a set of lightweight, swipe-aware delivery policies designed to reduce unnecessary data transfers. Section V concludes the paper by summarizing our main findings and discussing current limitations. Finally, Section VI outlines directions for future work, with a focus on improving delivery strategies for short-form video platforms.

[1] https://github.com/JamilAL01/YouTube-Shorts-Dataset

II. Related Work

Quality of Experience (QoE), data transfer efficiency, and protocol behavior are central topics in video streaming research. In recent years, sustainability and energy efficiency have also received increased attention [18], [14], [19]. In this section, we focus on prior work related to adaptive streaming, data efficiency, and user interaction, as these aspects are directly relevant to network performance and user experience.

Several studies have proposed techniques to improve the efficiency of mobile video consumption through context-aware adaptations. Liu et al. [20] and Isuwa et al. [21] investigated adaptive brightness control based on ambient light conditions to reduce energy consumption during video playback. Seeliger et al. [22] introduced AI-driven video encoding techniques to lower transmission overhead, while Turkkan et al. [23] explored reinforcement learning approaches for energy-efficient Adaptive Bitrate (ABR) selection. Herglotz et al. [15] analyzed viewport and resolution mismatches in mobile video streaming and showed their impact on data waste and perceived QoE.

From a networking perspective, most prior work has focused on traditional long-form video streaming, studying ABR performance under constrained or variable network conditions [24], [25]. These approaches typically assume continuous playback and relatively stable user engagement. However, such assumptions do not hold for short-form vertical video platforms, where user interactions are highly dynamic. Applications such as YouTube Shorts and TikTok are characterized by short viewing durations, and bursty traffic patterns [26], which pose new challenges for existing streaming strategies.

Although short-form video now represents a significant portion of mobile video traffic, its delivery characteristics are not fully addressed by conventional adaptive streaming solutions. To bridge this gap, several recent studies have begun to consider short-form consumption patterns. Works such as Dashlet [12] and related approaches [27] introduce delivery mechanisms that account for frequent swiping, while others propose adaptive preloading and bitrate selection strategies tailored to short videos [8], [17]. Additional efforts focus on reducing data waste through refined pre-buffering strategies while maintaining acceptable QoE [13], [17]. Despite these advances, existing studies remain limited in scale and often rely on simulations or partial system models, leaving application- and protocol-level behavior under real short-form viewing conditions insufficiently explored.

In contrast, our work provides a measurement-driven analysis of short-form video streaming at the segment level, focusing on buffering behavior and unnecessary data transfers caused by frequent swiping. By quantifying data waste under realistic viewing scenarios, we expose inefficiencies in current delivery mechanisms and highlight the limitations of conventional adaptive streaming designs in fast-paced, mobile-centric environments. Our findings offer empirical insights that complement existing work and inform the design of more efficient short-form video delivery systems.

III. Methodology

A. Scope and Case Study Selection

To investigate data consumption patterns in short-form video streaming, we focus on YouTube Shorts as a representative case. YouTube Shorts exemplify typical short-form behavior [28], [5], including rapid content turnover, algorithmic auto-play, and brief, swipe-driven user sessions. Our analysis aims to understand how content delivery mechanisms, prefetching strategies, buffering dynamics, and user interactions combine to create data inefficiency at the network level.

B. Data Collection

Accurate and comprehensive data collection is critical for characterizing YouTube Shorts content and user behavior. To capture a representative sample of current short-form video trends, we used the YouTube Data API[2] to gather metadata of a large set of short videos, including *video_URL, video_title, likes, comments, views, and category_name*. Recognizing that platform-specific trends evolve rapidly, we leveraged 50 trending TikTok hashtags [29] as search queries to ensure our dataset reflects actual user interests and viral content across platforms. This cross-platform approach helps capture popular videos relevant to the short-form ecosystem that might be missed using standard keyword searches.

To make the most of the YouTube Data API daily quota of $10\,000$ units, we used keyword searches. Each search costs 100 units and returns up to 50 videos. For every video, two extra requests for metadata and channel details cost 2 more units, giving a total of 4 units per video. Collecting about $100\,000$ videos required around $400\,000$ units, or about 40 days at the full daily limit, assuming no duplicate results. This approach helped us build a large dataset while respecting API limits.

The dataset consists of two parts. First, we collected metadata for over $100\,000$ short-form videos across multiple content categories[3], including video title, category, duration, and engagement metrics. Second, to support detailed analysis of playback behavior and data usage, we streamed the full videos using browser developer tools. Videos were streamed at multiple resolutions to examine encoding variations and adaptive bitrate behavior. From the full dataset, a representative subset of $10\,000$ videos was selected for swiping-based measurement experiments. This subset preserves the original distribution of content categories, durations, and popularity metrics (e.g., views, likes), ensuring diversity of user experience.

This comprehensive dataset enables us to correlate metadata attributes with actual streaming performance and resource consumption under realistic user interactions, including prefetching and buffering behavior.

[2]https://developers.google.com/youtube/v3
[3]https://gist.github.com/dgp/1b24bf2961521bd75d6c

During dataset construction, we encountered several challenges that required careful attention to maintain data integrity:

- **Filtering complexities:** Not all videos retrieved through '/shorts/' URLs are consistently treated as Shorts by YouTube. In some cases, these links redirect to standard '/watch?v=' videos. This behavior reflects YouTube's dynamic classification system, which labels a video as a "Short" based on factors such as aspect ratio and duration. To ensure only true short-form content was included in our dataset, additional filtering steps were applied.
- **Unavailable content handling:** Some videos became unavailable due to being set as private or removed by their creators. To maintain dataset quality, we implemented additional filtering by checking each video URL for accessibility, excluding those that returned errors or indicated private/unavailable status. Approximately 4.95% of videos were removed for these reasons, ensuring that the dataset contains only publicly accessible content.

These challenges led to imperfect query success rates and some inefficiencies in API unit consumption, extending the actual data collection period beyond the ideal minimum of 40 days. Not all API quota units were fully utilized or resulted in successful video retrievals.

Unlike previous datasets (e.g., [28]), our approach incorporates trending TikTok tags to capture real-time user interests, enhancing the representativeness of the collected videos. With this dataset, we can perform detailed analyses of video duration distributions, content popularity, and engagement metrics such as likes, comments, and views. These factors are crucial for modeling user interaction and evaluating streaming efficiency under realistic conditions.

Beyond metadata collection through the YouTube Data API, we downloaded the actual video files at all available resolutions using the tool yt-dlp [30]. Unlike API queries, this process retrieves the real video content, which is more time-consuming and bandwidth-intensive. The benefit of downloading videos at multiple resolutions is that it allows us to examine encoding variations and adaptive bitrate strategies, thereby providing insights that metadata alone cannot reveal. The dataset was carefully filtered and updated to ensure high relevance and accuracy over time.

C. User Behavior and Data Consumption

Short-form video platforms are characterized by rapid and impulsive interactions, where users often swipe videos after only a few seconds of playback. This behavior can lead to substantial volumes of downloaded data that are never watched, resulting in avoidable bandwidth and energy waste.

Our goal is to quantify the relationship between user behavior and network-level data usage by computing the *data wasted* metric:

$$\text{Data Wasted} = \text{Data Downloaded} - \text{Data Watched} \tag{1}$$

Experimental Setup and Tools. We designed a hybrid system combining browser automation, network logging, and video processing to accurately reconstruct segment-level behavior during streaming sessions:

- *YouTube IFrame API* [31]: embeds each Short video in a controlled web interface for consistent playback.
- *JavaScript & Puppeteer* [32]: simulates viewing scenarios and logs all segment requests via browser DevTools[4].
- *yt-dlp* [30]: downloads video and audio streams by *itag* for offline byte-level and encoding analysis.
- *ffmpeg* [33]: slices media files into time-aligned segments and calculates playback duration per byte-range.

Viewing Scenarios Simulated. In this work, we focus on YouTube Shorts, originally limited to videos of 60 s or less under the platform's initial short-form definition[5].

To evaluate streaming efficiency under diverse engagement patterns, we simulate five representative user interaction scenarios inspired by prior behavioral studies on short video consumption [12], [17]. These studies indicate that users often swipe videos either shortly after playback begins or near the end. Accordingly, we define the following scenarios:

- *Skip Directly*: user swipes 0.5 s after video starts.
- *Skip at 5s*: user watches 5 seconds before swiping.
- *Skip at 10s*: user watches 10 seconds before swiping.
- *Skip at 20s*: user watches 20 seconds before swiping.
- *Full View*: the entire video is played.

Each scenario allows us to evaluate how early downloading of video segments affects data waste and streaming efficiency under different user engagement patterns.

Playback and Segment Logging. We use a custom web interface to stream a representative subset of 10 000 YouTube Shorts from our dataset via the IFrame API (see Table I). This sample size was chosen because streaming the full dataset of 100 000 videos was impractical due to the considerable time required. Preliminary experiments on subsets of 1,000 and 5,000 videos indicated convergence in the observed metrics, supporting the representativeness of the chosen sample.

Video playback was orchestrated using Puppeteer [6] across predefined playback scenarios that emulate typical user interactions (e.g., continuous viewing and swiping between videos). During playback, logs from browser DevTools were captured to extract detailed segment request information.

Each segment is represented by its byte-range, timestamp, and associated `itag`[7]. For each video, we collect all video and audio segment metadata across considered streaming scenarios, associating them with a unique video ID for downstream analysis and processing.

[4]https://developer.chrome.com/docs/devtools

[5]As of October 2024, YouTube Shorts supports videos up to 180 s in length. However, our dataset and analysis are based on content published prior to this update.

[6]Puppeteer is a Node.js library that provides a high-level API to control Chromium-based browsers programmatically, enabling automated web interaction and measurement.

[7]`itag` is a stream identifier used by YouTube to specify encoding parameters for a video or audio stream, such as resolution, codec, and bitrate. Each `itag` corresponds to a unique format: https://gist.github.com/MartinEesmaa/2f4b261cb90a47e9c41ba115a011a4aa

Accurate Segment Duration Extraction. A critical step in analyzing the relationship between playback behavior and data usage is accurately determining segment durations. DevTools logs do not provide explicit temporal durations for each video segment, nor do the manifest files (MPD or M3U8) always yield precise timing. To overcome this, we developed a systematic method to estimate segment durations.

Precise Segment Duration Matching. To obtain accurate durations for both video and audio streams, we employ a byte-level matching methodology:

1) Extract all segment byte-ranges and `itag` values for video or audio-only streams from DevTools logs.
2) Download the corresponding video/audio stream for the same `itag` using `yt-dlp`.
3) Slice the downloaded stream into byte-aligned chunks matching the DevTools byte ranges using `ffmpeg`.
4) Measure the exact playback duration of each chunk via decoding, enabling precise per-segment time estimation.

This method applies to both audio and video streams, providing high accuracy across all segments [8].

Figure 1 summarizes the workflow we designed to extract accurate segment durations for both video and audio streams. This process ensures precise matching between downloaded byte ranges and actual playback time coverage.

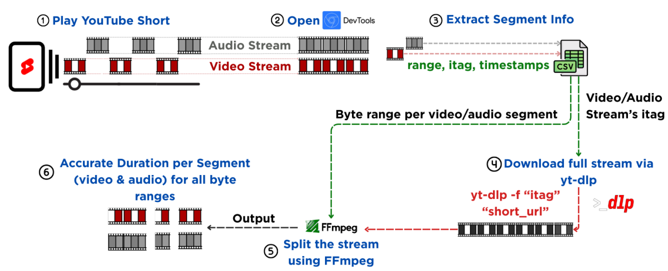

Figure 1: Pipeline to derive accurate segment durations using DevTools, yt-dlp, and FFmpeg.

Challenges and Refinements. Initially, we split segments from one endpoint to the next (e.g., from 2 s to 4 s), but this caused data loss due to intra-frame positioning in video codecs[9]. Segments that did not start at keyframes resulted in approximately 10% data loss.

To resolve this issue, we modify our method to always split from 0 s to X s, then compute differences between cumulative parts to reconstruct precise segment boundaries.

Achieving byte-level precision with `ffmpeg` required splitting with small time steps (0.1 s) and tuning encoding parameters to disable optimization features such as re-indexing, ensuring accurate playback and data alignment.

Foundations for Scenario-Based Waste Estimation. Building on our data collection and segment analysis pipeline,

[8]The total size of all DevTools segments matches the downloaded media file size within a 5 KB margin, confirming the validity of this approach.

[9]Intra-frame keyframes are complete image frames used as reference points in video compression. Decoding requires starting from a keyframe to ensure correct rendering. Segments beginning mid-GOP (Group of Pictures) may lack reference data, leading to partial data loss.

Content category	# Videos	Avg. views	Duration [Min-Max] (Avg, s)	Resolution range	Avg. bitrate 360p (Mbps)↑
Nonprofits & Activism*	97	1.74M	[8.0 - 161.0] (avg: 34.04)	360 p - 360 p	**1.316**
Gaming*	540	3.55M	[4.0 - 129.0] (avg: 24.68)	360 p - 720 p	**1.277**
Science & Technology*	341	4.39M	[3.0 - 178.0] (avg: 30.65)	144 p - 720 p	**1.171**
Comedy*	370	12.37M	[4.0 - 127.0] (avg: 28.50)	360 p - 720 p	**1.047**
Sports*	550	12.74M	[4.0 - 137.0] (avg: 23.01)	360 p - 1080 p	**0.976**
Entertainment*	2221	6.71M	[5.0 - 178.0] (avg: 29.08)	240 p - 720 p	**0.946**
News & Politics*	108	1.98M	[5.0 - 165.0] (avg: 33.64)	360 p - 720 p	**0.935**
Pets & Animals	172	5.28M	[5.0 - 75.0] (avg: 24.03)	360 p - 720 p	0.934
Cars & Vehicles	152	3.88M	[5.0 - 60.0] (avg: 21.84)	360 p - 720 p	0.909
Music	194	4.20M	[5.0 - 125.0] (avg: 26.48)	240 p - 720 p	0.736
Film & Animation	328	4.39M	[4.0 - 155.0] (avg: 32.05)	360 p - 720 p	0.670
Education	588	1.11M	[3.0 - 131.0] (avg: 29.46)	360 p - 720 p	0.493
How-to & Style	791	8.21M	[5.0 - 180.0] (avg: 28.37)	144 p - 720 p	0.490
People & Blogs	3383	3.41M	[3.0 - 180.0] (avg: 25.83)	360 p - 1440 p	0.483
Travel & Events	165	0.27M	[6.0 - 61.0] (avg: 22.03)	360 p - 360 p	0.455
Total / Avg	**10000**	**4.95M**	**[3.0 - 180.0] (avg: 27.51)**	144 p - 1440 p	**0.85**

Table I: Summary of YouTube Shorts categories in our dataset, including number of videos, average views, duration, resolution range, and average bitrate at 360 p. Categories marked with an asterisk (*) were selected for deeper analysis.

we evaluated data waste across five representative viewing scenarios: immediate skip (0.5 s), skip after 5 s, skip after 10 s, skip after 20 s, and full view.

For each video in our dataset, Puppeteer [32] was used to simulate these behaviors, while browser DevTools captured all requested segments. Using the `itag` codes, the corresponding media streams were downloaded via `yt-dlp` [30], and `ffmpeg` was employed to extract precise playback durations.

This allowed us to compute, for each scenario, the difference between total data downloaded and actually watched, quantifying the data wasted due to user interactions. These measurements form the foundation for analyzing the efficiency of short-form video streaming platforms, such as YouTube Shorts, under diverse engagement patterns.

Buffer Behavior Analysis. To assess YouTube Shorts' buffering behavior, we instrumented the playback interface to capture segment requests during video sessions. For each video, we recorded the timing, duration, and size of buffered audio and video segments ahead of the current playback point. This allowed us to characterize the platform's prefetching strategy and evaluate how much content is buffered relative to actual viewing. Our observations indicate that YouTube Shorts' buffering strategy is not fully optimized for the rapid, impulsive viewing typical of short-form videos.

Bitrate Distribution and VBR Encoding. Downloaded video segments were analyzed using `ffmpeg` [33] to extract segment-level bitrate and size information. The analysis confirmed that YouTube Shorts content is encoded with Variable Bitrate (VBR), where high-motion segments have larger byte sizes and higher bitrates, while static or low-motion segments are more compressed. This variability underscores the need for precise byte-level matching, as time-based estimation alone would misrepresent actual data transferred.

D. Prefetching and Download Policies

To evaluate strategies for reducing unnecessary data transfers in short-form video streaming, we designed a set of controlled download policies that operate at the segment level. All policies were implemented on top of the dash.js framework [10], which provides fine-grained control over segment scheduling, buffering behavior, and adaptive bitrate decisions in DASH-based players. Using dash.js allowed us to emulate alternative download behaviors in a fully controlled environment while preserving standard streaming mechanisms.

The policies target different sources of data waste observed in our measurements, including aggressive prefetching, excessive buffer filling, and large initial segment downloads. Specifically, we consider the following policies:

- **Policy A – Controlled Prefetching:** Limits prefetching of the *next* video by downloading only a small portion of its first segment. This policy aims to reduce wasted data caused by speculative downloads of videos that users are unlikely to watch, while leaving the buffering behavior of the currently playing video unchanged.
- **Policy B – Low-Bitrate First Segment:** Reduces the bitrate of the first segment of the current video while keeping the original segment duration unchanged. The selected bitrate is not the lowest available quality, but a conservative value based on the average bitrate of the available representations. This reduces the data downloaded at startup without changing segment timing or buffer targets, and may indirectly influence later downloads by affecting early download dynamics.
- **Policy C – Buffer Size Control:** Restricts the maximum buffered data for the current video by applying a safe buffer limit. By limiting how many segments can be downloaded ahead of playback, this policy prevents excessive pre-buffering when users swipe away early, which we identify as a major source of data waste.

In addition to evaluating each policy separately, we also analyze their combined effect to assess cumulative data savings. Unless otherwise stated, all experiments use the same viewing

[10]https://github.com/Dash-Industry-Forum/dash.js

scenarios defined earlier and are run under identical, controlled network conditions, ensuring a fair comparison between the baseline and each policy configuration.

IV. Results and Findings

This section presents the main findings from our analysis of YouTube Shorts streaming behavior across different watch durations and segment characteristics. We examine, for each video only, the relationship between data downloaded and data wasted, as well as buffering, prefetching, and segment-level download dynamics during partial views. Quantitative metrics and visualizations are used to highlight inefficiencies in current short-form video delivery.

A. Impact of Swiping on Data Consumption and Waste

Unlike traditional long-form video streaming, where users engage with a single video for extended periods, short-form platforms (e.g., YouTube Shorts, TikTok, Instagram Reels) promote rapid swiping and short attention spans. As a result, many videos are only partially viewed or skipped within seconds. To evaluate streaming efficiency under these interaction patterns, we simulate watch-time scenarios ranging from immediate swipes (e.g., $0.5\,\mathrm{s}$) to full video playback.

Our analysis quantifies the gap between downloaded and actually viewed data, highlighting wasted data. Figure 2 shows that when users swipe quickly (e.g., after 5 or $10\,\mathrm{s}$), over 60–90% of prefetched data remains unused due to aggressive preloading strategies.

At a larger scale, this behavior has significant implications. Global Internet traffic is estimated at around 4.8 ZB/year according to Cisco [34], and up to 10–12 ZB/year in more recent estimates [35]. With video accounting for about 70% of this traffic and short-form content representing roughly 80% of mobile video [36], short-form videos can be conservatively estimated to account for around 50% of total traffic.

Given our observed waste under short viewing conditions, (60–90%, ~70% on average), this suggests that roughly 35% of global Internet traffic may be effectively unused, corresponding to approximately 1.6–4 ZB of wasted data annually. Using an energy intensity of 0.05 kWh/GB [37], this translates to about 80–200 TWh of electricity per year, comparable to the output of roughly 10 to 25 nuclear power plants.

This finding suggests that adaptive buffering strategies based on user behavior, swipe prediction, or video popularity could reduce unnecessary bandwidth and energy use, contributing to more sustainable streaming systems.

B. Data Waste Across YouTube Content Categories

To study category-specific streaming behavior, we selected 7 representative categories from 15 YouTube Shorts categories in our dataset. Table I summarizes all categories, with the selected subset marked by an asterisk (*). These seven were chosen from 10 000 sampled videos based on high average bitrates at 360 p. For each, we report the number of videos, average duration, resolution range, views, and bitrate.

We randomly selected up to 50 videos per category to ensure balanced comparisons. For each video, we measured total data

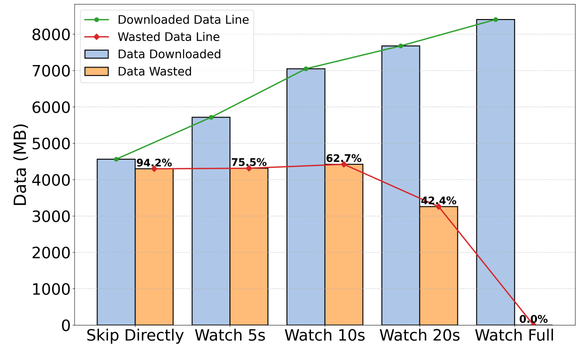

Figure 2: Total data downloaded vs. data wasted across swiping scenarios (with % represents the percentage waste).

downloaded and the portion wasted under simulated viewing scenarios. This highlights how content type affects streaming efficiency, where engagement varies across categories.

Figure 3 shows clear differences in average data waste across categories, influenced by motion intensity, encoding complexity, video duration, and resolution. Scenario 3 (skip at $10\,\mathrm{s}$) is used for detailed comparison. Most categories have average waste between $0.7\,\mathrm{MB}$ and $0.9\,\mathrm{MB}$ per video, while *Nonprofits & Activism* and *Science & Technology* peak around $1.1\,\mathrm{MB}$. Categories like *Sports* and *Gaming* show lower waste, likely due to shorter durations or higher engagement.

Figure 4 illustrates per-video variability within each category, showing that averages can hide substantial differences among individual videos. These results suggest that content-aware streaming strategies, adapting prefetching to category characteristics, could further reduce data waste.

C. Buffer Size Evolution Over Time

Streaming protocols often prefetch multiple segments to ensure smooth playback. While effective for long-form videos, this can lead to over-buffering in short-form contexts, where users frequently swipe before a video ends.

Figure 5 shows segment requests for five fully watched videos of the same length. YouTube Shorts downloads each video quickly, often before halfway, and prefetches the next one, typically 20–40% of its duration. This “Next-One” strategy [38], [13] enables smooth transitions but leads to data waste when users skip early.

Figure 6 shows buffer size evolution over time for a single representative video at multiple resolutions. While buffered duration remains similar across resolutions, higher resolutions fetch more data, increasing redundancy when users swipe early. This example is illustrative, but it suggests that time-based buffering can lead to inefficiencies in swipe contexts.

D. Bitrate Variability in Short-Form Videos

Short-form video platforms like YouTube Shorts use content-aware encoding strategies, resulting in Variable Bitrate (VBR) streams where segment sizes fluctuate with scene complexity, motion, and visual detail. Unlike Constant Bitrate

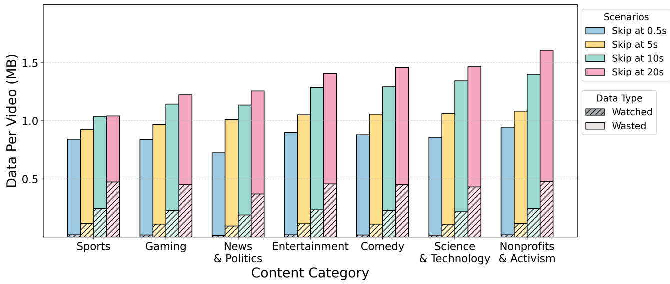

Figure 3: Data watched and wasted per YouTube Shorts category across scenarios.

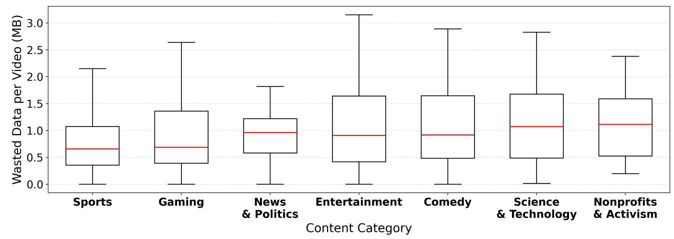

Figure 4: Distribution of wasted data per video for Scenario 3 (skip at 10 s). Boxes show interquartile range, whiskers show full range, and the red line marks the median.

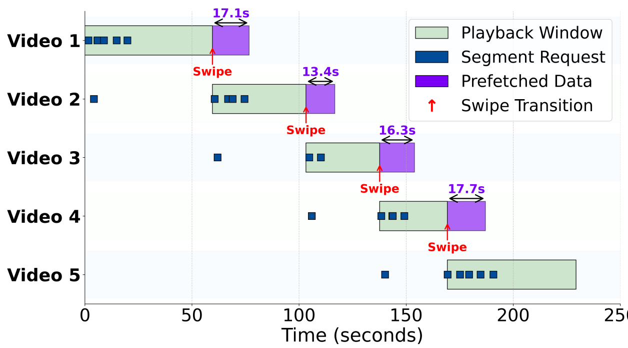

Figure 5: Segment requests and prefetched data when streaming short-form videos on YouTube Shorts.

(CBR) encoding, which maintains uniform rates, VBR adapts to content to balance quality and compression efficiency.

This variability is especially noticeable in YouTube Shorts, where segment sizes can vary widely even within a single video. To illustrate this, we selected four categories: *Pets & Animals*, *Gaming*, *Entertainment*, and *How-to & Style*, which differ in scene complexity, bitrate, and video duration. For example, *Gaming* videos often have high bitrates due to fast motion, while *How-to & Style* videos are simpler and lower-bitrate. *Entertainment* videos tend to be longer, while *Pets & Animals* clips are shorter (see Table I).

Figure 7 shows that high-motion categories like *Gaming* and *Pets & Animals* exhibit greater bitrate volatility than more static categories like *Entertainment* or *How-to & Style*.[11]

We also observe that video bitrates are front-loaded, with early segments having higher bitrates. This can create a quality bias at the start and cause buffering inefficiencies: if users swipe away quickly, significant data in high-bitrate early segments is wasted. Because short-form videos involve frequent swipes and brief viewing, prefetching these early segments may lead to substantial data waste, up to 90% in immediate-swipe scenarios.

[11]To isolate content effects, we selected the same number of videos per category, all encoded at 720 p and 20 s duration.

E. Download Percentage Across Playback Scenarios

We analyze the fraction of each video that is downloaded under varying playback scenarios, where users watch only a portion of the video before swiping. This helps quantify the mismatch between data downloaded and actual user engagement, which is especially pronounced in platforms dominated by rapid, impulsive swiping.

Figure 8 shows the average fraction of video data downloaded across different video durations and segment counts. We observe an early burst of downloads: regardless of the video's total length, most videos start downloading at least 2–3 segments within the first 2 s of playback. This behavior reflects aggressive pre-buffering aimed at minimizing startup stalls and ensuring smooth playback.

Our analysis shows that up to 90% of downloaded data can be wasted when users swipe away without watching the video. This occurs because high-bitrate segments are downloaded immediately, often within the first 2 s of playback, regardless of total video length. These findings highlight the inefficiency of current prefetching strategies and emphasize the potential benefits of adaptive prefetching that better aligns downloads with actual user behavior.

F. Impact of Segment-Level Policies on Data Waste

We now evaluate the impact of the segment-level policies introduced in Section III, controlling next-video prefetching (Policy A), first-segment bitrate (Policy B), and buffer size (Policy C), on data waste across different viewing scenarios.

Scenario	Policy A	Policy B	Policy C	All
Skip Directly	14.96	22.43	86.08	96.69
Watch 5s	1.48	8.18	61.52	61.11
Watch 10s	-2.52	3.65	54.75	45.51
Watch 20s	-2.94	3.94	50.68	48.90

Table II: Data savings % per policy across user scenarios.

All policies were evaluated under the same viewing scenarios defined earlier (0.5 s, 5 s, 10 s, and 20 s), using a controlled *dash.js*-based environment. Table II reports the percentage of data savings relative to the baseline behavior.

Figure 9 illustrates these results visually. Several important observations emerge.

First, **Policy A** achieves only modest savings in the skip-directly scenario and can even introduce slight overhead for

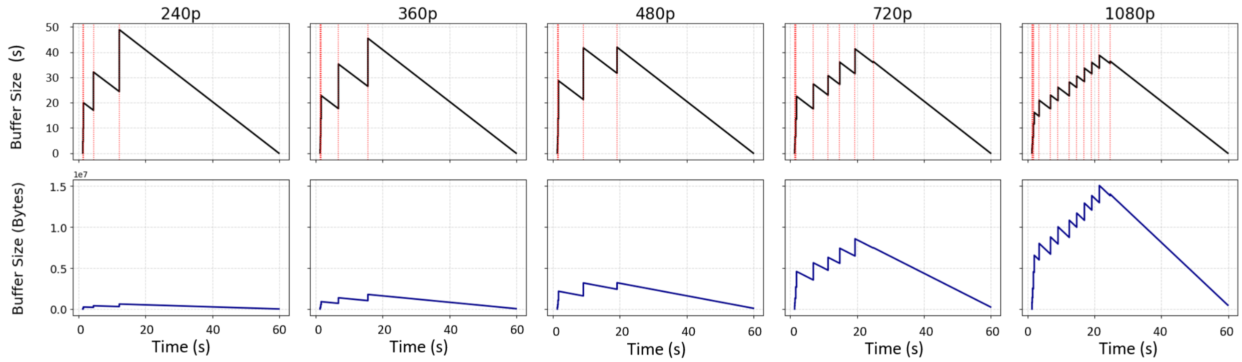

Figure 6: Buffer evolution for a single video across resolutions (240 p–1080 p) in seconds (top) and bytes (bottom).

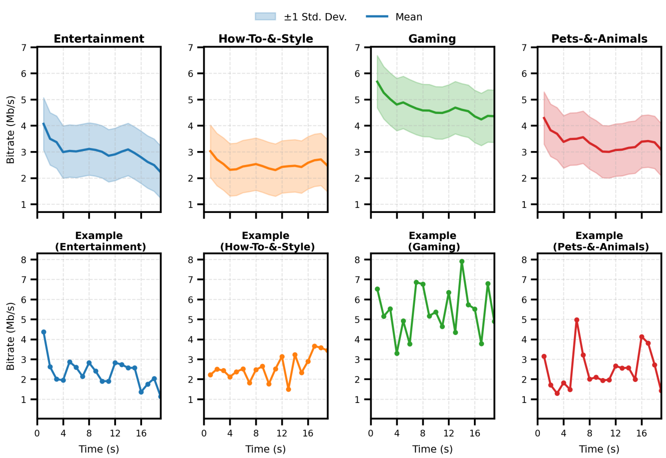

Figure 7: Avg bitrate and individual video bitrate by category.

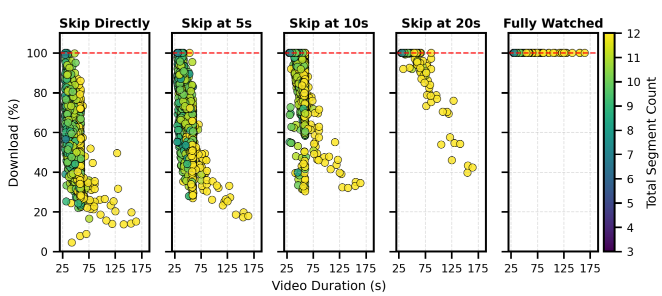

Figure 8: Download percentage across playback scenarios for varying video durations and segment counts.

longer partial views (10s and 20s). This behavior is explained by the fact that, even when users swipe almost immediately, a non-negligible amount of data is downloaded beyond the first segment due to minimum buffer requirements and in-flight requests. Since Policy A does not constrain this additional downloaded data, its effectiveness is inherently limited.

Second, **Policy B** provides moderate but stable savings across scenarios. Although it primarily targets the first segment, reducing its bitrate can indirectly reduce additional downloaded data by shortening the startup phase and limiting

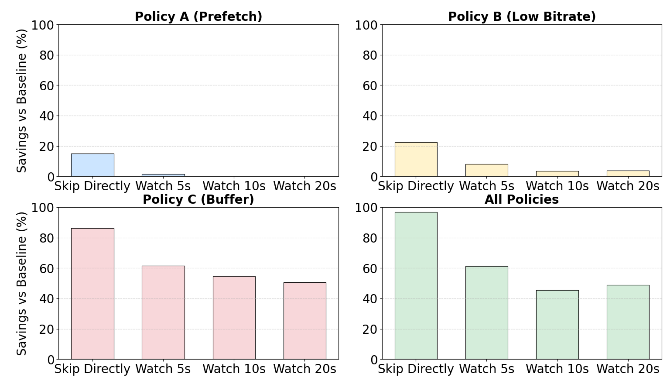

Figure 9: Percentage reduction in wasted data relative to the baseline for different segment-level policies.

the time window during which subsequent segments can be fetched before the user swipes.

Third, **Policy C** consistently delivers the largest reduction in data waste across all scenarios. By directly bounding the buffer size, this policy limits the amount of data that can be downloaded before a swipe occurs. As a result, it substantially reduces waste in early-swipe scenarios and remains effective even for longer partial views, although its relative benefit naturally decreases as more content is watched.

When all policies are combined, cumulative savings are substantial, particularly for early-swipe scenarios where unnecessary downloads dominate. Interestingly, savings for the 20 s scenario exceed those for 10 s. This occurs because longer partial views trigger more aggressive buffering in the baseline, creating greater opportunity for the combined policies to prevent unwatched downloads. Overall, these results demonstrate that controlling buffer growth is the most effective lever for reducing data waste, while prefetch and bitrate-based policies provide complementary benefits.

V. Conclusion

This work analyzes data waste in short-form video streaming using YouTube Shorts, based on measurements from over 10 000 videos. We find that aggressive prefetching and rapid user swiping lead to severe inefficiencies: up to 90% of downloaded data is never viewed, and more than 60% of data can be wasted within the first 10 s of playback.

Data waste varies by content category, video length, and segment-level buffering behavior, with longer and higher-bitrate videos (*Nonprofits & Activism*, *Science & Technology*) exhibiting the highest waste. High-bitrate early segments increase the gap between delivered and consumed data.

We show that simple segment-level controls on first-segment bitrate and prefetching can reduce unnecessary downloads by up to 80–90%, demonstrating that policy-based approaches can significantly improve efficiency in short-form streaming.

VI. Future Work

Future work will explore adaptive prefetching and buffering based on predicted user engagement and content features to reduce data waste while preserving QoE. We also plan to evaluate these methods in real network conditions and extend them to other short-form platforms.

Acknowledgment

This work has been supported by the French government National Research Agency (ANR) through the UCA JEDI (ANR-15-IDEX-01) and EUR DS4H (ANR-17-EURE-004) and by the France 2030 program under grant agreements No. (ANR-23-PECL-0003 and ANR-22-PEFT-0002).

References

[1] Cisco, "Cisco annual internet report (2018–2023) white paper," 2020.

[2] Ericsson, "Ericsson mobility report, june 2023," 2023.

[3] T. Van Daele, A. Iyer, Y. Zhang, J. C. Derry, M. Huh, and A. Pavel, "Making short-form videos accessible with hierarchical video summaries," in *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pp. 1–17, 2024.

[4] DataIntelo, "Short video platforms market report." https://dataintelo.com/report/short-video-platforms-market.

[5] YouTube Press, "Youtube official press site." https://blog.youtube/press/.

[6] "Youtube shorts statistics 2025 (top picks)." https://www.demandsage.com/youtube-shorts-statistics/. Accessed: 2025-07-07.

[7] Neal Mohan, "Neal Mohan at cannes lions 2025: What 20 years of YouTube reveals about creativity's future," june 2025.

[8] N. T. Phong, T. T. Huong, P. N. Nam, T. C. Thang, and D. Nguyen, "Joint preloading and bitrate adaptation for short video streaming," *IEEE Access*, vol. 11, pp. 121064–121076, 2023.

[9] Q. Zhou and Y. Wang, "Swiping more, thinking less: Using tiktok hinders analytic thinking," *Cyberpsychology: Journal of Psychosocial Research on Cyberspace*, vol. 17, no. 2, 2023.

[10] K. Y. Tam and M. Inzlicht, "Fast-forward to boredom: How switching behavior on digital media makes people more bored.," *Journal of Experimental Psychology: General*, vol. 153, no. 10, p. 2409, 2024.

[11] X. Tian, X. Bi, and H. Chen, "How short-form video features influence addiction behavior? empirical research from the opponent process theory perspective," *Information Technology & People*, pp. 387–408, 2023.

[12] Z. Li, Y. Xie, R. Netravali, and K. Jamieson, "Dashlet: Taming swipe uncertainty for robust short video streaming," in *Proceedings of the 20th USENIX Symposium on NSDI*, pp. 1583–1599, 2023.

[13] C. Tran, T. Nguyen Duc, N. Bach, P. Tan, and E. Kamioka, "Sprinkle prebuffer strategy to improve quality of experience with less data wastage in short-form video streaming," 2022.

[14] C. Müller, R. Seeliger, S. Steglich, and S. Arbanowski, "Towards sustainable video streaming: Evaluation of ai approaches for content aware encoding," in *2023 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pp. 1–4, IEEE, 2023.

[15] C. Herglotz, W. Robitza, M. Kränzler, A. Kaup, and A. Raake, "Modeling of energy consumption and streaming video qoe using a crowdsourcing dataset," in *2022 14th International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–6, IEEE, 2022.

[16] W. Robitza, A. M. Dethof, S. Göring, A. Raake, A. Beyer, and T. Polzehl, "Are you still watching? streaming video quality and engagement assessment in the crowd," in *2020 twelfth international conference on quality of multimedia experience (QoMEX)*, pp. 1–6, IEEE, 2020.

[17] C. Zhou, Y. Ban, Y. Zhao, L. Guo, and B. Yu, "Pdas: Probability-driven adaptive streaming for short video," in *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 7021–7025, 2022.

[18] C. Herglotz, W. Robitza, A. Raake, T. Hossfeld, and A. Kaup, "Power reduction opportunities on end-user devices in quality-steady video streaming," in *2023 15th International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 79–82, IEEE, 2023.

[19] M. U. Khan, S. Abbas, S. U.-J. Lee, and A. Abbas, "Measuring power consumption in mobile devices for energy sustainable app development: A comparative study and challenges," *Sustainable Computing: Informatics and Systems*, vol. 31, p. 100589, 2021.

[20] D. Liu, C. Qian, H. Rong, S. Zhou, C. Xiang, and H. Jiang, "Energy and qoe optimization for mobile video streaming with adaptive brightness scaling," *ACM Transactions on Sensor Networks*, pp. 1–24, 2024.

[21] S. Isuwa, D. Amos, A. K. Singh, B. M. Al-Hashimi, and G. V. Merrett, "Maximising mobile user experience through self-adaptive content-and ambient-aware display brightness scaling," *Journal of Systems Architecture*, vol. 145, p. 103023, 2023.

[22] R. Seeliger, C. Müller, and S. Arbanowski, "Green streaming through utilization of ai-based content aware encoding," in *2022 IEEE International Conference on IoTaIS*, pp. 43–49, IEEE, 2022.

[23] B. O. Turkkan, T. Dai, A. Raman, T. Kosar, C. Chen, M. F. Bulut, J. Zola, and D. Sow, "Greenabr: Energy-aware adaptive bitrate streaming with deep reinforcement learning," in *Proceedings of the 13th ACM Multimedia Systems Conference*, pp. 150–163, 2022.

[24] J. Zhang, Z.-J. Wang, S. Guo, D. Yang, G. Fang, C. Peng, and M. Guo, "Power consumption analysis of video streaming in 4g lte networks," *Wireless Networks*, vol. 24, pp. 3083–3098, 2018.

[25] F. Tashtarian, D. Lorenzi, H. Amirpour, S. Afzal, and C. Timmerer, "Coconut: Content consumption energy measurement dataset for adaptive video streaming," in *Proceedings of the 15th ACM Multimedia Systems Conference*, pp. 346–352, 2024.

[26] Y. Liu and F. Liu, "Study on the use and satisfaction of short video app for college students—based on the empirical survey of hubei college students," in *Proceedings of ICMIR 2020*, pp. 199–207, Springer, 2021.

[27] S. Zhu, T. Karagioules, E. Halepovic, A. Mohammed, and A. D. Striegel, "Swipe along: a measurement study of short video services," in *Proceedings of the 13th ACM Multimedia Systems Conference*, 2022.

[28] C. Violot, T. Elmas, I. Bilogrevic, and M. Humbert, "Shorts vs. regular videos on youtube: a comparative analysis of user engagement and content creation trends," in *Proceedings of the 16th ACM Web Science Conference*, pp. 213–223, 2024.

[29] TikTok Ads Creative Center, "Popular hashtags on tiktok." https://ads.tiktok.com/business/creativecenter/inspiration/popular/hashtag/pc/en, 2025. Accessed: 2025-06-20.

[30] yt-dlp Contributors, "yt-dlp: A feature-rich command-line audio/video downloader." https://github.com/yt-dlp/yt-dlp, 2025.

[31] Google Developers, "Youtube iframe player api reference." https://developers.google.com/youtube/iframe_api_reference, 2025.

[32] Google Developers, "Puppeteer documentation." https://pptr.dev/, 2025.

[33] B. Pugh, "Trim videos without re-encoding." https://www.brandonpugh.com/til/video/trim-without-reencoding/, 2023. Accessed: June 2025.

[34] Cisco, "Cisco predicts more ip traffic in the next five years than in the history of the internet." https://newsroom.cisco.com/press-release-content?articleId=1955935, 2017.

[35] Anything Counter, "Internet traffic per second." https://anythingcounter.com/internet-traffic-per-second?, 2024.

[36] Ericsson, "Short-form video dominates mobile traffic." https://www.ericsson.com/en/reports-and-papers/mobility-report/articles/short-form-dominates-video-traffic, 2025.

[37] J. Aslan, K. Mayers, J. G. Koomey, and C. France, "Electricity intensity of internet data transmission: Untangling the estimates," *Journal of Industrial Ecology*, vol. 22, no. 4, pp. 785–798, 2018.

[38] J. He, M. Hu, Y. Zhou, and D. Wu, "Liveclip: Towards intelligent mobile short-form video streaming with deep reinforcement learning," in *Proceedings of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, pp. 54–59, 2020.