HAL open science

Modeling Rabbit-Holes on YouTube

Erwan Le Merrer, Gilles Trédan, Ali Yesilkanat

▶ **To cite this version:**

Erwan Le Merrer, Gilles Trédan, Ali Yesilkanat. Modeling Rabbit-Holes on YouTube. 2022. hal-03620039v1

HAL Id: hal-03620039

https://hal.science/hal-03620039v1

Preprint submitted on 25 Mar 2022 (v1), last revised 19 Jul 2023 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Surfing Personalization for Quantifying the Rabbit Hole Phenomenon on YouTube

ERWAN LE MERRER, GILLES TRÉDAN

Numerous discussions have advocated the presence of a so called *rabbit-hole* (RH) phenomenon on social media, interested in advanced personalization to their users. This phenomenon is loosely understood as a collapse of mainstream recommendations, in favor of ultra personalized ones that lock users into narrow and specialized feeds. Yet quantitative studies are often ignoring personalization, are of limited scale, and rely on manual tagging to track this collapse. This precludes a precise understanding of the phenomenon based on reproducible observations, and thus the continuous audits of platforms.

In this paper, we first tackle the scale issue by proposing a user-sided *bot*-centric approach that enables large scale data collection, through autoplay walks on recommendations. We then propose a simple theory that explains the appearance of these RHs. While this theory is a simplifying viewpoint on a complex and planet-wide phenomenon, it carries multiple advantages: it can be analytically modelled, and provides a general yet rigorous definition of RHs. We define them as an interplay between *i)* user interaction with personalization and *ii)* the strength of attraction of certain video categories, which cause users to quickly step apart of mainstream recommendations made to fresh user profiles.

We illustrate these concepts by highlighting some RHs found after collecting more than 16 million personalized recommendations on YouTube. A final validation step compares our automatically-identified RHs against manually-identified RHs from a previous research work. Together, those results pave the way for large scale and automated audits of the RH effect in recommendation systems.

1 INTRODUCTION

Modern recommendation systems, and in particular personalization on video platforms, are tuned to maximize users watch time [2] and their presence on platforms. This has been associated with major societal impacts [23], triggering a debate on whether or not the personalization brings users on *radicalization paths* [8, 19]. To a less extreme end, the problem of users getting a narrow view on the possibilities offered by the platforms have been studied qualitatively under the terms of *filter bubbles* [15, 26, 27], and more recently the *rabbit hole* (shorthanded RH through the paper) phenomenon [5, 8, 13, 21].

The dynamics of the personalization by these recommendation systems is based on opaque factors, and their operating platforms appear as black boxes to their users [16]. Yet the current understanding is that the RH effect can be defined as the situation where "recommendations influence viewers of radical content to watch more similar content than they would otherwise" [8]. This stresses a dynamic where the watch history of users is a crucial preamble to future recommendations, but also that there is an average case where these users would not access these videos otherwise (*i.e.*, most probably would view more mainstream content). Such definitions are certainly hard to leverage when it comes to practical quantification, for instance when auditing such a platform. While tools for measuring simpler phenomenons such as popularity of videos with regards to geography [1], or the amount of ads [24] for instance emerged, such tools are lacking for quantifying the RH effect. Despite its importance for the rising field of algorithm auditing [17], quantification is inherently a challenge, as it requires to quantify "average" experiences from a collection of experiences made unique by the very process of personalization. This paper thus seeks to articulate a precise definition of the RH effect so that reproducible and automated means of measure are made possible (rather than relying on manual tagging [8, 19]).

We take the step to collect personalization to multiple and controlled user profiles in YouTube, in order to match a theory we propose under the form of a model. This model is composed by the two core components at stake in the

previous definition [8]: there is *i)* a *feedback loop* in the viewing patterns of users, for *ii)* specific types of videos that have the characteristic to be far away from a "standard" usage. In this paper we coin these videos *attractors*, and we will characterize them by their capability to collapse the entropy of more mainstream video recommendations. The feedback loops for personalization to a user will be modeled and measured by using the *autoplay*-recommendations [11, 12, 22] provided by YouTube to that user.

Contributions. This paper makes the following main contributions. *i)* We present in Section 3 an experimental setup that allows to **measure recommender personalization through bots** and validate this experimental setup on a large collected corpus. This is key to automate the study of the trapping dynamics of users into RHs. *ii)* We propose a simple **model to explain the formation of RHs**. This model permits to detect users entering RHs by relative comparison (Section 4). Finally, *iii)* we validate our model on the personalized data collected by bots, quantify the attractivity of some channel categories, show that we can **automatically identify the RHs** that were manually labeled by a previous research work, and discuss a measure of the personnalisation strength of any video.

2 BACKGROUND

The lack for a precise understanding of the filter bubble or RH effect. Defining a RH is no easy task. In [8], RH is implicitly defined as filter bubbles where: "recommendations influence viewers of radical content to watch more similar content than they would otherwise". In [5], the authors (although not precisely defining it so) relate their finding to communities in the network formed by recommended channels. Both papers point to work in [13], which relies on clustering (precisely, non-negative matrix factorization) to study how "*recommendation* can have the undesirable consequence of a user being excluded from information that is not aligned with their existing perspective, potentially leading to immersion within an ideological bubble". All above references point to Pariser's definition of filter bubbles [15]: "Together, [recommendation engines] create a unique universe of information for each of us—what I've come to call a filter bubble—which fundamentally alters the way we encounter ideas and information". Interestingly, Pariser insists in the individualized nature of this bubble ("You're alone in it"), while more recent works rather focus on the social effect of such bubbles (*e.g.*, considered as a pathway to conspiracy theories). These rather insist on the dangers of community wide bubbles that would attract many users into specific recommendation regimes. While these seminal works all refer to the same phenomenon, by defining RHs through their consequences, they leverage subjective notions (ideology, radicalization) that are hard to quantify. Yet, we believe that a rigorous quantification of the RH phenomenon is essential for a continuous and automated surveillance.

Personalization is key to the phenomenon, yet difficult to study. Personalization leverages both the user information and content information to provide tailored recommendations [9]. This is obviously the core mechanism to study when one wants to assess the impact of the recommendation algorithm onto users consumption. Unsurprisingly though, personalization is complex to collect and study at scale, as multiple profiles must be tracked. This is why state of the art papers on understanding filter bubbles or RHs solely consider *contextual* recommendation, which is not adapted to users but a generic proposal related to the consumed item [14, 19]. The most recent approach by Ribeiro et al. [19] for instance collects the channels recommended by other selected channels, as general and static contextual references. Roth et al. collect contextual recommendations around certain seed video categories [20]. In [8], recommendations made to a single anonymous profile are collected; this profile does not simulate any video watch, precluding personalization; human labelling of videos is also performed. Hussein et al. [4] nevertheless managed to setup a collection by bots (through the scriptable Selenium browser), yet leveraged human annotators to label videos (debunking, neutral, promoting) for

a study on misinformation prevalent in YouTube searches and recommendations. These works clearly establish the existence of RHs in contextual recommendations; however, we here argue that personalization is a key element to consider and study in order to explain the current experience of users with regards to RHs.

3 SURFING PERSONALIZATION ON YOUTUBE USING BOTS

3.1 Bots for collecting personalization at scale

We rely on the use automated scripts (*i.e.*, *bots*) in order to simulate human users watching and interacting with videos on YouTube. Our bot actions are implemented using the scriptable Puppeteer headless Chrome browser[1]. The most important actions we have implemented for this study are: the search for a given term in the search bar, the mouse click on a recommendation based on its position in the list, the collection of videos from a given user channel, the viewing of a video (and the associated collection of recommendations), and the access to the YouTube welcome page to gather context-free recommendations before and after searches and video views by a bot.

To control personalization as much as possible, and avoid measurement artifacts due to browser fingerprinting for instance, we took the following steps. The bots are built within Docker containers that are instantiated and then destroyed after each user simulated interaction with YouTube, so that we are sure to remove all traces and variables between multiple runs. This is in particular the case for cookies, that are thus reinitialized every time. At the run of a new bot for a particular action and data collection, we randomize variables such as the browser user agent, and announced screen resolution. The watch time for videos by the bots is randomized to a mean of 5 minutes ±10% (if videos last at least this duration of course)[2].

We summarize and depict the instrumentation of this study in Appendix Section A. We report a lightweight operational execution of each bot, that permits its execution on basic machines such as on the four Intel NUC computer architectures we used in parallel for this study.

Data collection summary. Our automated data collection campaign took place from February 18th to March 22nd, 2022. The recommendations were captured on the YouTube welcome page, by a scroll-down fetch resulting in around 300 recommendations gathered each time, before and after each video watched by the bot.

We summarize these statistics in Table 1. For the experiments concerning our contribution *i)*, bots are viewing in sequence 20 videos from kid channels, or from the welcome page as a control experiment. These walks collected personalized recommendations, that were in turn automatically labeled by the tool we will introduce in the sequel. We note that each walk duration is in the order of 100 minutes, since bot "watch" 5 minutes of each video along their five hops walk. For the experiments concerning our contribution *ii)*, bots started from various channels pertaining to several well separated categories that will be discussed (and taken from [19], and from famous channels for kids), and from a neutral profile as a control experiment. Walks duration are then of length around 25 minutes. This sums up to $16,386,099$ personalized recommendations collected.

From recommendations to similarities between users. This study relies on bots following so called *Autoplay recommendations* (as studied in related works *e.g.*, [11, 12, 22]). When watching a video, the corresponding autoplay recommendation is the top recommendation displayed (right hand side of the YouTube video page), and is the next

[1]https://github.com/puppeteer/puppeteer – We note that recommendations (be they contextual or personalized) are only available at the time of our collection campaign using a Javascript enabled browser: the mere use of a fast yet basic static web scrapper (such as *e.g.*, scrappy) does not allow recommendation collection, as it was the case 5 years ago [7].

[2]This value is way over the 30 seconds known to be the limit of YouTube to consider a real video watch and then to increment its view counter, see *e.g.*, https://growtraffic.com/blog/2017/08/youtube-video-counts-view.

	Personalization to bots (Sec. 3.2)	RHs vs channel categories (Sec. 5)
launched bots (walks)	533	5, 438
length of walks (hops)	20	5
recom. collected	5, 746, 494	10, 639, 605
auto. labelling resolved	5, 726, 296	N/A

Table 1. Statistics on the collected data. Bots viewing videos and scraping the resulting YouTube personalization.

video to be played if no other action is taken by a user. It certainly indicates a high confidence of the recommendation algorithm that it is suited to the viewer.

The principal substrate we are thus building on for this study is personalization to simulated user profiles, under a *vector* representation. This vector space corresponds to the *one-hot encoding* of each of the encountered video IDs. For each profile, we thus associate a vector keyed by the video ID on YouTube (*e.g.*, `A5F9jOHbi5I`), containing the integer representing the amount of times this particular video has been recommended to the bot at a given step (0 if never)[3]. We call these *recommendation vectors*. All the vectors corresponding to our collection campaign are then perfect candidates to compute similarities between each of then, using for instance the *cosine similarity* (hereafter denoted S_c), well established in the field of recommender systems [10].

While bots collecting personalization and these vectors are sufficient to study the RH effect, we first introduce the labeller we leverage to validate the instrumentation approach.

An automatic labeller for testing personalization to bots: labelling videos in YouTube kids. A first salient question is whether or not a fully automated process as we depicted yields a reaction (*i.e.*, personalization) by the YouTube recommendation algorithm.

For this validation step, we rely on an automated way to label collected recommendations (as opposed to human tagging in *e.g.*, [4, 8, 19]), in order to observe if YouTube recommendations are increasingly composed of the category of videos viewed by the bots. If so, YouTube reacts to the bots actions as expected, and thus automation is a way to observe recommendations based on input actions that we can script.

YouTube kids[4] is the alternative for kids to YouTube [25], targeting the following profiles: "preschool" are 4 years old or less, "younger" are from 5 to 7, and "older" are from 8 to 12. We will generally call a *kid* user one that belongs to one of these three categories. Yet, while YouTube is encouraging kids to use YouTube kids instead, all major content creators for kids (counting hundreds of millions of views, as reported on Table 2) are present on YouTube.

We found out that videos share the same video ID on both YouTube and YouTube kids. Moreover, videos provided by YouTube kids contain an age tag in the associated page metadata. We leverage this information as an age tracker in the following way. Using the ID of any video encountered in YouTube recommendations, we query YouTube kids. As a result, either we obtain the video page from YouTube kids, in which case we know it is a "kid" video and we can extract its precise age label from the page tag, either we obtain an error page, in which case we label it "adult video" since YouTube did not make this video available for kids. We report that only 533 videos out of 5, 726, 296 failed to be resolved in this dataset.

Using such a labeller for automatic labelling is convenient, as opposed to the tedious task of manual labelling; we now leverage it for observing the personalization proposed to our bots.

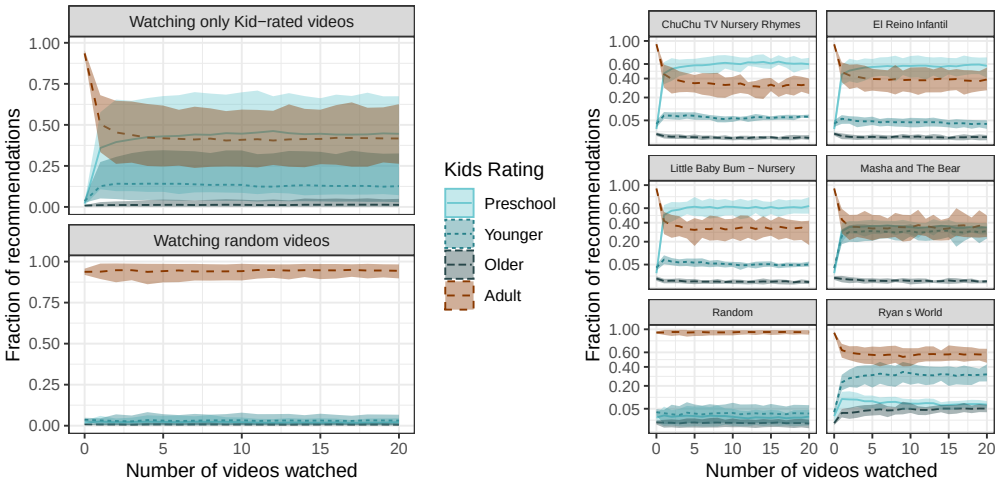

Fig. 1. The impact of bots actions on personalization: (*left*) Evolution of the fraction of recommended kid rated videos, when watching either random or either kid videos. (*right*) A detailed view by kid channels (Random baseline included for reference). The shaded area delimits the 10 and 90 percentile.

3.2 Measuring personalization on kids profiles

To validate our YouTube instrumentation, we measure the ability of our bots to trigger personalized recommendations that depend on their past video consumption. Thanks to the labeller, we can test the following hypothesis: "bots watching children content will get an increase in recommended children content".

Our approach is as follows: given a list of famous children video channels (that we list in Table 2), we emulate fresh profiles that watch only videos from these channels. After each video watched, we collect recommendations available on the bot welcome page. For comparison, a control experiment is realized by fresh profiles that watch random videos picked from their welcome page before any action is performed. All recommendations collected by both random and kid profiles are then automatically labeled. More specifically, we perform multiple runs where a bot starts from one of the five most accessed kids channels on YouTube, and watches in sequence 20 randomly selected videos from that channel.

Figure 1 presents the results of this experiment for which 5, 726, 296 recommendations were collected (51.3% from random profiles, 48.7% from kids profiles). The figure on the left presents the evolution of the recommendation composition (on the y-axis) for the kid profiles (top-figure) and the random baseline (bottom-figure) as a function of the number of watched videos (x-axis). We note that originally ($x = 0$), the watch history of all profiles being empty, videos for the two profiles get the same recommendation mix containing on average 93.7% of Adult (*i.e.*, non kid), 2% Preschool, 1.8% Younger and 0.2% Older videos. This proportion remains unchanged when watching random videos, which is the expected behaviour of our control condition. At contrary, recommendations to kid profiles quickly adapt to converge to a stable mix containing only 41.5% of Adult videos and 44.7% Preschool, 12.7% Younger and 1.2% Older. In other words, as compared to the baseline the ratio of non adult videos has been multiplied by more than 9.2. We draw two main conclusions from this experiment: *i)* Our experimental setup effectively manage to trigger personalization from

[3]Hence, we leave aside the position of a given video in the recommendation list. Although this definitely constitutes a valuable information, weighting the relative importance of videos being recommended first or last is difficult and left for future works.
[4]https://www.youtubekids.com

Channel type	Channel name	#views	#videos
Kids (K)	Ryan's World	48,975,229,688	2,091
Kids (K)	El Reino Infantil	44,074,006,603	1,157
Kids (K)	ChuChu TV Nursery Rhymes & Kids Songs	35,098,632,207	452
Kids (K)	Little Baby Bum - Nursery Rhymes & Kids Songs	34,570,591,653	1,669
Kids (K)	Masha and the bear	17,643,829,073	857
Alt-Right (AR)	Black Pigeon Speaks	40,774,438	175
Alt-Right (AR)	The Golden One	12,286,875	605
Alt-Right (AR)	NeatoBurrito Productions	8,334,309	320
Alt-Right (AR)	Australian Realist	5,472,094	290
Alt-Right (AR)	Prince of Zimbabwe	40,088,933	20
Alt-light (AL)	StevenCrowder	1,443,559,041	1,296
Alt-light (AL)	Rebel News	628,496,085	16,535
Alt-light (AL)	Paul Joseph Watson	517,887,497	829
Alt-light (AL)	Mark Dice	440,494,445	1,522
Alt-light (AL)	Sargon of Akkad	306,059,074	1,187
Intellectual dark web (IDW)	PowerfulJRE	1,704,287,921	600
Intellectual dark web (IDW)	JRE Clips	4,421,584,838	5,393
Intellectual dark web (IDW)	PragerU	1,439,923,494	1,755
Intellectual dark web (IDW)	The daily Wire	763,331,055	6,940
Intellectual dark web (IDW)	The Rubin Report	419,351,645	3,109
Media (M)	Vox	2,734,974,347	1,357
Media (M)	GQ	2,565,984,292	1,846
Media (M)	VICE News	2,364,198,449	5,843
Media (M)	WIRED	2,565,984,292	3,308
Media (M)	Vanity Fair	1,711,181,309	2,430

Table 2. Statistics on the channels used as video pools and autoplay walk starts in this study. In this paper we bring the channels for kids (grey rows); these are selected after a top-view ranking[5]. The four remaining categories are taken from Ribeiro et al. [19].

the recommendation algorithm. And *ii)* this personalization is triggered quickly: the recommendation mix changes drastically within the first three watched videos, and further consumed videos have little to no impact.

Figure 1-(right) is identically constructed, but provides a detailed view at each kid channel profile. Note the square-rooted y-axis scale. While this view confirms the previous conclusions, one can further observe that different kid channels trigger different personalizations. Channels providing Preschool content (*e.g.*, Little Baby Bum) have the highest impact on diminishing adult recommendations, while channels targeting older kids (*e.g.*, Ryan's World) trigger less Preschool recommendations. Interestingly, no channel manages to substantially increase the amount of Older-type recommendations. While providing a definite explanation would require deeper investigations, we believe this is mostly due to Older-type videos being quite rare compared to the other kid types, perhaps because it is difficult to draw a separation line between Older-type and Adult-type videos.

Summary: Our bot instrumentation manages to trigger YouTube personalization. This personalization, in the case of kid profiles, quickly adapts, and impacts a significant fraction of the provided recommendations.

4 A MODEL FOR THE RABBIT HOLE EFFECT

In the previous section, we observed the impact of watched videos (*i.e.*, history) on recommendations. Since this observed personalization is the 'raison d'être' of many recommenders, it does not alone justify the existence of RHs. In this section, we thus investigate the trapping dynamics of users into RHs. More precisely, we elaborate a minimalist model that emulates them. Through this model, we argue that the formation of RHs relies on two ingredients: i) the user interplay with personalization, and ii) the presence of video sets we call *attractors*.

4.1 Feedback loops and attractors

A simple model. Let us first consider a simple recommender R. Suppose the set of videos is containing only two types of videos: A and B-types. R works as follows: for every user, R remembers the 10 last watched videos. When issuing recommendations, R recommends a B-type video with a probability p_B proportional to the number of B-type videos already in history. For instance, if a user has $5/10$ B-type videos in her history, recommendations will be of type B with

probability $p_B = 1/2$. Once R has selected the video type (A or B), it picks the video to recommend uniformly at random among videos of that type. R thus captures the following simple intended property: imagine A are Adult videos, and B kid videos; the more kid video are watched, the more kid videos are recommended.

Users always follow recommendations by watching a uniformly selected random video among all recommended ones[6]. This essential component of the model creates a feedback loop between R and the user: the more she watches B the more she gets, which, if she follows recommendations, will get her to watch more B videos and so on and so forth.

To sum up, this model has essentially two key factors: *i)* a *recommender* R, which users always leverage by following recommendations, closing the *feedback loop* between recommendation and history: together, R and users form a dynamical system. *ii)* A strong distinction between video types: videos types are either A type, either B type. Moreover, this type distinction is associated with a straightforward reaction from R: B type video consumption systematically increase the likelihood of B type recommendation. As a reference to their key role in the recommendation dynamical system, we refer to B as an *attractor*. This resulting model is a discrete dynamical system that we can analyse and simulate. Please refer to Appendix B for a more formal description of the model. To maintain a high level perspective, we now focus on simulation results.

Simulations using this model. We simulate a system of $n = 100$ users iteratively obtaining $y = 50$ recommendations among a catalog V of size $|V| = 1000$ videos. We consider such a set of $|B| = 100$ specific videos ($|A| = 900$ remain in the general category). Users' history is of size $h = 10$ and is initially filled with videos picked uniformly randomly (*i.e.*, belong to B with probability $|V|/|B|$).

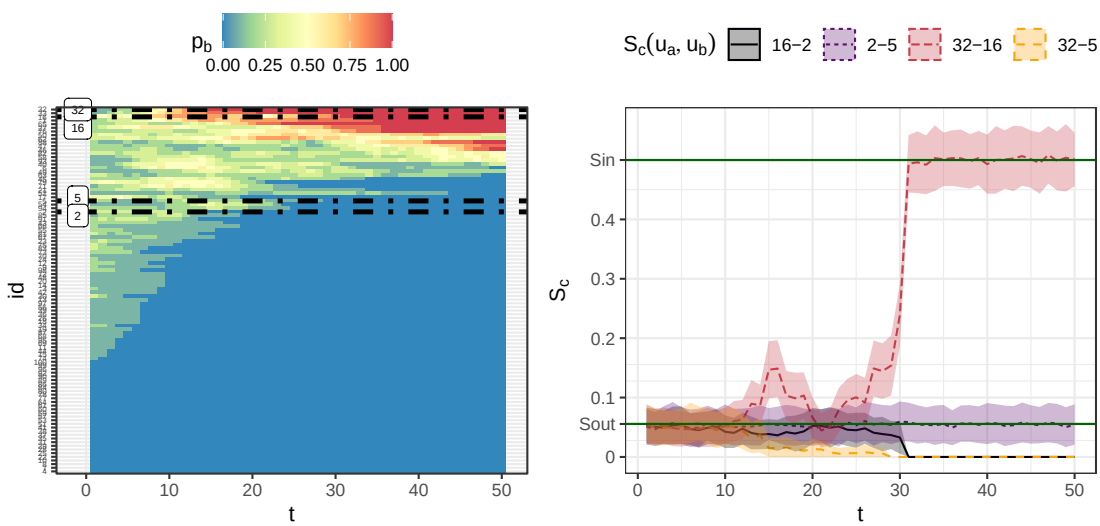

Fig. 2. Simulations of the model: *(left)* Evolution of p_b along rounds (x axis) for all users (ordered by average p_b, y-axis). When $p_b = 0$ (yellow), users never get recommended any video from B (the RH); when $p_b = 1$ (blue), users always get recommended B videos. Dashed lines highlight users showcased in the Figure on the right. *(right)* Recommendation (cosine) similarities for four particular users: divergence emerges between a user in the RH (user 32) and a user getting mainstream recommendations (user 5). Recommendation similarity is high between both users in the RH (32 and 16). Value Sin (resp. $Sout$) represents the theoretical similarity value, given by the model, between two users in (resp. out) of the RH.

[6]We note that the prevalence of clicks on recommendations over searches is major, around 70% in 2018 as indicated by YouTube product chief, please refer to https://www.cnet.com/news/youtube-ces-2018-neal-mohan/.

Figure 2 *(left)* simulates for each user 50 rounds of picking a video according to the model (x-axis represents time in number of rounds). Each user (on y-axis) corresponds to a horizontal line, with the color indicating the probability p_B to get recommended a video from B (the darker the most likely). For instance a yellow dot indicates that the given user on the y-axis has her history do not contain a single video from B, which turns out to provoke a probability p_b of 0 to get recommended a video from B. Initially, some users have p_b values different than 0 or 1, while a significant number (31) of users already start with a p_b equal to zero[7]. As the time progresses, more and more users fall in either one of the following states: $p_b = 0$, users never get recommended any B-type video, and $p_b = 1$, users always get recommended B-type videos. In other words, users for which $p_b = 1$ are trapped in the B-type RH. Before being locked in either state ($p_b \in \{0, 1\}$, users in the transient states are recommended both A and B videos. Although the figure shows only the first $t \leq 50$ rounds, the system ultimately converges to a state where all users are either only recommended B videos, either never recommended B videos.

Of course, in vivo users would not get recommended only videos from two solely categories, even when converged (this is often referred to as *serendipity* [6]). This model can be trivially extended to adapt the specifics of particular setups. For instance, while we here focus on the analysis of a single RH (B), results are consistent in the case of multiple RHs (as confirmed empirically by simulations we made). On the analytical side, this model can be treated as an absorbing Markov Chain (which states capture all possible history states) to leverage the powerful analysis toolbox available in this domain. User might as well choose videos on their own. However, while this model definitely constitutes an extreme case, we believe it is sufficient to illustrate the natural phenomenon behind the trapping dynamics into RHs: if recommenders suggest videos similar to past watched videos, and if users watch recommended videos, such a feedback loop naturally emerges. These feedback loops turn tight categories (either A, either B) of the recommender turn into attractors, trapping users into a RH.

Summary: We simulate a recommender's feedback loop: B-type videos are recommended to a user proportionally to this user's past consumption of B, while this user follows these recommendations. Simulations illustrate the trapping of users in a RH: the system quickly converges to a state where users either only either never see B videos. The necessary ingredients are a recommender that users are interacting with by following recommendations, and attractor videos.

4.2 Detecting rabbit holes from the user side

Interestingly, this model also sheds light on black box observation scenarios (*i.e.*, from the perspective of users being recommended videos). We focus on one question: "can users detect they entered a RH?", to which we answer positively.

Modeling user detection of RHs. After several videos at least have been consumed by every user, these users belong to three categories: users in U_a never get recommended videos from B ($p_b(u) = 0$), users in U_b always get recommended videos from B ($p_b(u) = 1$), and finally users in U_{ab} that still get recommendations from both A and B categories.

Let u, u' two users, and their collected recommendations Y, Y'. Since we defined B to be a small subset of the whole video catalog, two U_b users will likely have similar recommendations; in other words $Y \cap Y'$ is large. The same is also true for U_a users, but to a lesser extent due to the larger set of videos in $A = V \setminus B$ and since $|V| >> |B|$. Finally, note that two users in different partitions will observe no intersection in their recommendations ($Y \cap Y' = \emptyset$)[8].

[7]Which meets the theoretical expectation: the probability of not picking a B video is $(1 - |V|/|B|)$, hence the probability of not picking a video h times is $(1 - |V|/|B|)^h \simeq 35\%$ chances given the selected parameters.

[8]Those observations also hold for the multiple RH case: a collection of categories $\{B_i\}_i$, provided those are tight $\forall i, j, B_i \cap B_j = \emptyset$ and small compared to the overall catalog: $\forall i, |V| >> |B_i|$. In such setting, users in different RHs will also have an empty recommendation intersection.

We further analyze this model in Appendix C, from which we extract the following results: given two users **in** the RH, their expected cosine similarity is $Sout = \frac{y}{|B|}$, and given two users **not** in the RH, their expected similarity is $Sin = \frac{y}{|V|-|B|}$, with $y = |Y|$ the number of recommendations returned by the platform. Since B-type videos only represent a small part of the whole video catalog V, those results confirm the intuition: users in the same RH will get a high similarity, hence RH can be detected by exploiting recommendation similarity. As this high similarity is synonym of a low diversity, we refer to this term as recommendation *entropy collapse*.

Simulations. This is confirmed by the numerical simulations that are presented Figure 2 *(right)*. It presents the evolution of the cosine similarity of four selected user couples. Users IDs 32 and 16 correspond to the two top dashed lines on the left plot, and correspond to users ending up in U_b (that is, they both end in the rabbit hole). Their recommendation similarity, originally similar to others, increases sharply to around $y = 0.5$. In contrast, users 5 and 2 correspond to the second pair of dashed lines on the left plot, are users ending up in U_a (*i.e.*, not in the rabbit hole). Their recommendation similarity stays low and rather stable across time around 0.06. Finally, the similarity between heterogeneous couples (namely $16 - 2$, $32 - 5$) drops to zero.

In other words, the recommendation similarity between two U_b users will be high, whereas it will be low for two U_a users, and zero between a U_b and a U_a user. Thus, by computing recommendations similarity, we can distinguish U_b users from others and identify B videos.

Summary: Our model captures the relation between users position with respect to a RH, and their recommendation vector similarity. This relation enables user-sided RH detection.

5 COLLECTED DATA AND THE RH EFFECT

We now confront the data we collected with the proposed model, to conclude on the possibility to observe the RH effect with our automated approach.

5.1 Down the RH through specific channel types?

We choose to leverage the channel types identified by Ribeiro et al. in [19]. They identified four of them on YouTube as channel providers, namely the alt-right (AR), alt-light (AL), media (M), intellectual dark web (IDW). We selected for each of them the five most viewed channels, that were still open at the time of our collection campaign (we for instance note that the most viewed channel for the AR category –named James Allsup– was closed). To these four, we added another arguably well separated type, the content providers for kids (K) we considered in the previous section. Table 2 lists the characteristics of these channels. We stress that *these labeled channels are used solely as a validation purpose* for our approach for detecting RHs, and that such external labelling is not required to perform the detection of RHs.

Each run by a bot proceeds as follows. A bot first randomly selects one channel among the ones we just listed. It then selects a video at random among all videos in that channel. This constitutes the starting video for its walk on the autoplay recommendations.

Experiment results. We performed 5,438 autoplay walks, each of length five from the various starting video channels. In Figure 3 are plotted the cosine similarities between the collected recommendation vectors at the end of each walk. It is based on the average blend of recommendations collected by each autoplay walk after its journey. For instance a given walk ID on the x-axis is compared to another walk ID on the y-axis: the clearer the color the more similar are

these two walks in the recommendations they receive (the diagonal is omitted for clarity: if $x = y$ then we have by construction a cosine similarity of 1). We grouped the channels by blocs following their types along both axes.

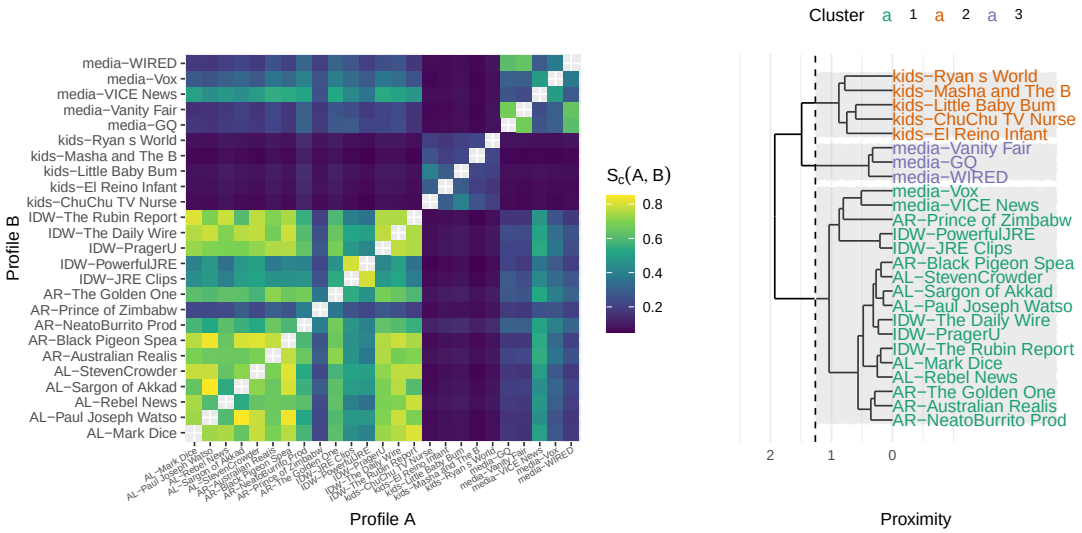

Fig. 3. (*Left*) The similarity of recommendations after autoplay walks starting from specific channels, sorted by categories on both axes. Blocs appear, indicating the consistency of recommendations received by bots depending on their starting channels. (*Right*) A dendrogram representation of the similarity between the channels, and the appearance of category-coherent clusters.

A first important observation is the heatmap's overall bloc structure that closely matches the different investigated channel types. For instance, the media channels cause a bloc in the upper right part of the figure, which means that a walk starting from these media channels tends to get recommended videos that are similar to any other walk from starting from media channels, compared to any other walk starting from other channel types, say *e.g.*, the kids channels.

The inter bloc comparison reveals that the most dissimilar recommendations are made for the kids and AR, AL and IDW categories, as one can expect (this is to be observed by the dark cross appearing in the figure). Similarities are high between the three AR, AL and IDW categories, as compared for instance to these three and the media category.

If we focus on the similarity within each bloc, we observe that walks starting from within the AR, AL and IDW channels are the closer in the recommendations they receive. This "internal" similarity is lower within the media bloc, and even lower within the kids bloc. We note an outlier with the Prince of Zimbabwe channel, tagged as AR by [19], yet appearing in our measurement closer to the media channels. Its last video was uploaded two years ago.

The pairwise channel similarities of this heatmap can be exploited through a hierarchical clustering. Using Ward's method leads to the dendrogram presented in Figure 3 (*right*). Cutting this dendrogram reveals coherent clusters of channel categories. However, the threshold (proximity) one considers for cutting, symbolized as a dashed line on the Figure, leads to different clusters; this is studied in the next Subsection.

5.2 Retrieving communities through clustering personalization to bots

While Figure 4 presents an inter-profile distance heatmap, it does not precisely define the boundaries of the potential RH into which walks are attracted. We now have a look at this aspect using clustering.

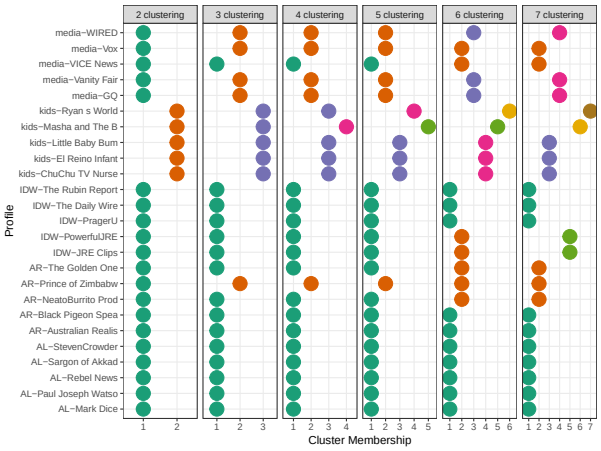

Fig. 4. Identifying rabbit holes automatically using autoplay walks from channels on the y-axis: relying on k-Means. k varies from 2 to 7, and resulting memberships (x-axis).

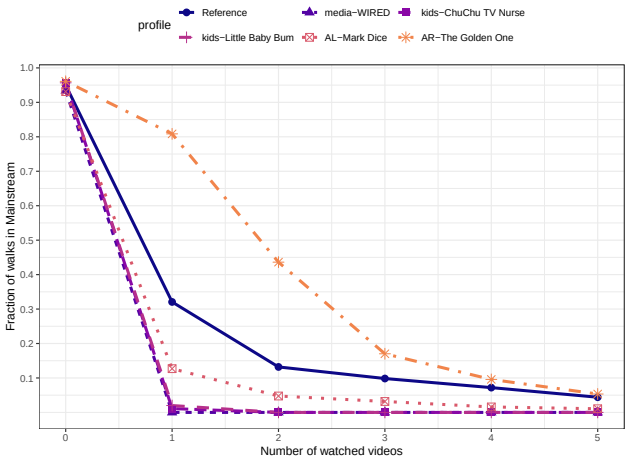

Fig. 5. Strength of attraction of several starting videos (from some K, M, AL and AR channels from Table 2).

We choose to focus on arguably the simplest clustering method, k-Means, which also has the advantage of being efficient to process the 16 million recommendations collected for this experiment. k-Means requires as input k, the number of clusters to be found. Figure 4 presents the identified clusters memberships: each vignette represents the clustering results for a given value of k, ranging from 2 to 7, which allows to observe the various resolutions at which RHs could be defined.

Partitioning the data into $k = 2$ clusters splits the profiles into two broad categories: kids profiles (cluster 2-orange dots) and adult profiles (cluster1-green dots). The $k = 3$-clustering corresponds to the macro categories we defined for our investigations in Table 2: kids, media, and right-wing profiles, with the notable exceptions of channel "Prince of Zimbabwe" that gets assigned to the media cluster, and Vice news assigned to the right-wing cluster. Further increasing the number k of clusters further refines the categories: $k = 5$ splits the kids category, $k = 6$ splits the right-wing category.

To complement these qualitative assessment of the identified RH, one can rely on several metrics defined to measure the quality of a clustering. We choose to focus on 3 metrics; their evolution is depicted in Table 3 as a function of the number of identified clusters. The first one is a standard unsupervised metric that is based on the ratio between intra-cluster variance and inter-cluster variance, the intuition being that an ideal clustering is composed of very homogeneous clusters (low internal variance) that are all very different from each other (high inter-cluster variance). This measure steadily increases as the number of cluster grows – as expected since higher k values provide more freedom for k-Means to optimize this metric.

In addition to this unsupervised metric, we can also leverage the categories assigned in [19] as well as our kid category, and use a supervised metric to measure to which extend the clusters match those categories. We rely for this on the Rand index [18], that captures the percentage of correct assignments issued by the clustering. It is maximized for $k = 6$ clusters, reaching a 77% agreement. Interestingly, if one chooses to merge all right-wing categories into a single category, one obtains a maximum at $k = 3$ with 88% accuracy, which fits well the three broad studied categories (media, right-wing and kids). Moreover, this shows how this clustering captures with a reasonable accuracy the RHs tracked statically and without personalization by the work in [19], although our findings rather indicate the presence of a larger RH spanning the whole right-wing (AL, AR and IDW categories).

5.3 When is a user entering a rabbit hole?

In this last experiment, rather than seeking the characterization of RH, we seek to characterize the strength of the video or category attractors. In other words, instead of characterizing the destinations of the personalization made to bots as in the previous experiment, we measure the speed of the entropy collapse due to personalization associated with target videos. The benefits of this perspective is to circumvent scale problems inherent to the notion of RH (how large is a RH ? if 30% of my recommendations are not in a RH, am I experiencing a RH?) and profit from a well defined reference we name *mainstream recommendations*: recommendations made initially to the bots before they take any action. The intuition is that welcome page recommendations provided to new users without history ought to be somehow generic.

	groups	2	3	4	5	6	7
1	BetweenSS/TotalSS	0.20	0.31	0.41	0.49	0.56	**0.62**
2	Rand: Original Labels	0.52	0.71	0.70	0.69	**0.77**	0.76
3	Rand:Right Merged	0.76	**0.88**	0.87	0.86	0.77	0.76

Table 3. Clustering quality metrics for Figure 4, as a function of the number of clusters created by the k-means.

Concretely, given a set of profiles from which we perform multiple autoplay walks of depth k, we perform the following steps. As a first step, we consider only *initial recommendation vectors*, collected by bots using fresh cookies before watching any videos. We compute the average initial recommendation vector $\hat{Y_\emptyset}$, which can informally be considered as the "mainstream barycenter". Then we observe the distribution of similarities between any initial recommendation vector and this center $\hat{Y_\emptyset}$. This allows to define a similarity threshold σ above which a recommendation vector is considered as *in the mainstream*. In other words, we train a classifier to detect recommendation vectors in the mainstream. Appendix Section D details the finding of this σ ceil. Then, given any recommendation vector Y, if $S_c(Y, \hat{Y_\emptyset}) < \sigma$, we consider that the walk corresponding to Y has left the mainstream. In addition, for a given profile that is the set of multiple walks, we are then able to compute the proportion of such walks that left the mainstream.

Figure 5 presents the collected data with this perspective. It considers autoplay walks starting from several video channels presented in Table 2, and plots the fraction of these that left the mainstream as a function of the number of videos watched. For reference, we add a special profile coined "Reference", that corresponds to autoplay walks issued from videos picked randomly on the welcome page of a fresh bot.

We stress that we only control the start video, as autoplay embed the recommender system's logic, so as soon as the bot accesses the recommended video, the walk can quit –or not– the start category. The purpose of the experiment is thus to measure the strength of attraction of the starting video, and to observe how the rest of the walk performs with regards to the awaited loss of entropy.

Interestingly, autoplay walks starting at a kid video instantly leave the mainstream (*i.e.*, fractions of walks remaining in the mainstream is close to zero after the first video watched), so do the walks from the Wired channel. In comparison, only around 55% of the walks starting at "AR- The Golden One" channel videos left the mainstream after two videos watched, indicating that the attraction of these is much lower than the kids one, and also much lower than our reference walks. We conjecture that this is due to the recommendation algorithm possibly avoiding the autoplay proposition to extreme content systematically, leading walks to enter more classic recommendations in the followup. This would be consistent with the remark made in [19] noticing that *more than* 75% *of outgoing edges from all communities pointed towards the Other node*, that are videos outside the tracked radical channels.

Summary: Experiment results indicate that: *i)* the automated autoplay walks from well separated video channels end up reproducing a RH effect, as indicated by the model. Walks from pre-labeled channels by a previous research paper consistently end up being recommended the same items. *ii)* Clustering the autoplay walks allows to retrieve communities with high recommendation similarities. This showcases the use of the approach to automatically retrieve various communities implied in a RH effect. *iii)* The attractivity of videos varies drastically, sometimes counter intuitively (kids videos are stronger attractors than AR or AL videos in our collected data).

6 LIMITATIONS AND DISCUSSION

Simulating user behavior. We have automated some actions of users on YouTube, following two main scenarios: the multiple consumption of videos of a single channel (Section 3.2), or consumption of the Autoplay videos recommended by YouTube (Section 5). While we have shown that such an automation already allows for quantifying the RH effect, we believe that more advanced bot behaviors are still desirable (which quality of imitation can be measured through metrics [3]). Indeed, even better simulating humans could trigger different personalization behaviours overlooked by our approach. Yet, defining, implementing and analyzing the results produced by such human-realistic bot behaviours is challenging. We believe this increased complexity would conflict with the clarity, objectivity and simplicity required for the definition of a 'standard' measure as pursued in this paper.

A threshold to the rabbit hole. While our bot-driven study permits to automatically retrieve video categories that were before identified by manual labelling (Section 5.2), there remains a question of some thresholds to set. This is the case for the clustering performed in Figures 3 (right) and 4 and for the threshold to consider in Figure 5. An expert analysis of the results by our method is required to discriminate stable clusters in the former cases (for choosing a fix k), and to fix a threshold in the latter (as detailed in Appendix Section D). Nevertheless we believe these steps to be lightweight as compared to manual labelling of possibly millions of videos or channels. In addition, the threshold to decide when a user is entering a RH can be the prerogative of future auditors or regulators.

Are rabbit holes a technical artifact? While the general opinion often tends to focus on specific and extreme topics for the observation of RHs (*e.g.*, political radicalization [8, 19]), we have seen that our definition allows for the consideration of attractors videos or channels that are part of any category. Their strength of attraction (Figure 5) is to be evaluated without a priori, possibly yielding counter intuitive remarks: the alt-right (AR) videos cause a lesser entropy collapse of recommendations than kids ones. Our method is thus neutral when it comes to auditing, and able to measure from a user perspective the current dynamics of the formation of RHs around any target topic. We believe that those quantitative approaches, instead of competing against the necessary qualitative and/or subjective approaches required to study RH-related phenomena, rather provide an objective standpoint than can support those studies.

7 CONCLUSION

We presented in this paper an automated way to audit in vivo the personalization made to users by the YouTube recommender, with a focus on the RH effect. We have shown that bots actually trigger personalization. We proposed a model that helps approaching the problem of the dynamic of trapping users into RHs. As a control experiment, we have shown that such an automated audit, without relying on any manual collection nor tagging, permits to extract video categories that have been studied manually by the previous state of the art paper on the topic. This proposal is generic and a priori applicable to all recommenders, as it solely proceeds using the `url` of recommended items, and no

other specific information. We believe this is an important step in the automated and continuous audit of recommender systems.

Appendices

A AN OVERVIEW OF THE INSTRUMENTATION IN THIS STUDY

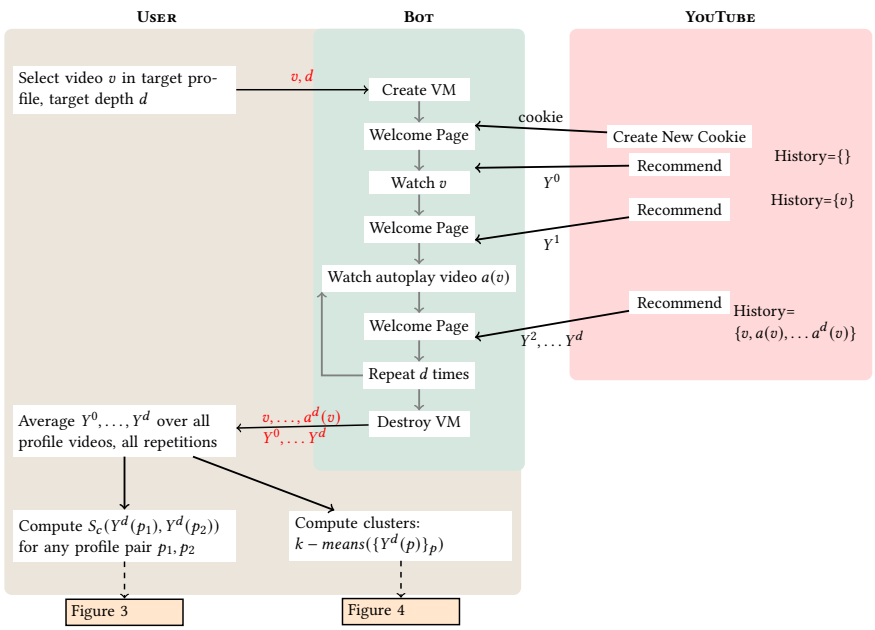

Fig. 6. Overview of the different steps performed in this study: our bots actions in YouTube, and the followup experiments.

B A MODEL FOR THE RABBIT-HOLE EFFECT

Let R be a recommender over a video catalog V. Let $B \subset V$ a subset of that constitutes, say, a specific video category (e.g. Kids, Right-wing, Portuguese Language). We note $A = V \setminus B$. Suppose R relies on a single feature in user profile "probability to recommend B video" p_b that is evaluated as the number of B videos in a history H of size h. Let $b = |B|, y = |Y|, v = |V|$. We assume users always pick a random video in recommendations Y.

Clustering for automated identification of rabbit holes. Concretely, to audit a set of videos $A = a_1 \dots a_k$, we generate a set of fresh users U, whose first action will be to watch a video from A. We then let those users enter autoplay walks. After this walk, we collect the recommendations associated with this profile $\mathcal{Y} = \{Y(u)\}_{u \in U}$. Finally, we partition this set $\mathcal{Y}$ into a collection S of k clusters: $S = \{s_1, \dots s_k\}$. While any clustering method might be used, one can leverage k-Means, which optimises: $arg\,min_S \sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{Y,Y' \in S_i} \|Y - Y'\|^2$. Since k-Means relies on random (centroids) initialization, it is considered good practice to conduct multiple runs and select the best obtained results. We followed the commonly prescribed value of 25 such restarts.

C USER-SIDED RABBIT-HOLE IDENTIFICATION

To estimate Cosine similarities $S_c(Y, Y')$ in our model, first recall that $S_c(Y, Y') = \frac{Y.Y'}{||Y||||Y'||} = \frac{|Y \cap Y'|}{\sqrt{y}.\sqrt{y}}$ since we deal with binary vectors in this simple model (*i.e.*, a given video appears at most once in a recommendation vector). Hence estimating the intersection of Y and Y' is sufficient to compute the recommendation similarity. Let $X = |Y \cap Y'|$ be the random variable capturing this value. Let us call $p_{u \mapsto u'}$ the probability that a video in Y is also present in Y': we have $X \sim \mathcal{B}(y, p_{u \mapsto u'})$ where $\mathcal{B}$ is the binomial law. We then split cases:

- $u, u' \in U_b$. In this case u has picked y videos among the b videos of B. Hence $p_{u \mapsto u'} = y/b$.
- $u, u' \in U_a$. In this case, videos are picked in $V \setminus B$. Hence $p_{u \mapsto u'} = y/(v - b)$.

Since the expected value of a binomial law $\mathcal{B}(n, p)$ is simply np, one can further develop $E[S_c(u, u')]$:

- $u, u' \in U_b$. Since $p_{u,u'} = y/b$, $E[X] = y^2/b$, and thus $E[S_c(u, u')] = \frac{y}{b}$.
- $u, u' \in U_a$. Since $p_{u \mapsto u'} = y/(v - b)$, $E[X] = y^2/(v - b)$, and thus $E[S_c(u, u')] = \frac{y}{(v-b)}$.

When applied to the system instance detailed above ($v = 1000, b = 100, y = 50$), one gets $E[S_c(u, u')|u, u' \in U_b] = 1/2$, and $E[S_c(u, u')|u, u' \in U_a] = 1/18$, which is validated by observations in Figure 2.

D ESTIMATING THE ATTRACTION STRENGTH OF VIDEOS

We here extend technical parts of Section 5.3. Let W a set of walks of length k, and let $\forall 0 \leq i < k, \forall w \in W, Y^i(w)$ be the recommendation vector collected by walk w after watching i videos.

First, we compute the mainstream recommendation vector as the mean of all recommendation vectors observed from any walk before its first watched video: $m_\emptyset = 1/|W| \sum_{w \in W} Y^0(w)$.

Yet, recommendation vectors are noisy by nature, hence to qualify "mainstream recommendation", we need to understand how this noise perturbs observations. To do so, we look at the similarity between recommendation vectors in the mainstream by construction (*i.e.*, no interactions nor videos watched by the bots) and pick a similarity threshold that includes the vast majority of those vectors in the mainstream. Figure 7 shows the density of this distribution, along with its evolution as videos are watched. We can observe a clear gap between 0 videos watched, and $i > 1$ videos watched. The gray vertical line represents the average similarity observed over the dataset (0.386). The dashed line represents the selected ceil : (σ =0.198). In other word, any walk Y such that $S_c(Y, m_\emptyset) > \sigma$ is classified as "in the mainstream". This ceil σ is set so that 95% of the Y^0 vectors are in the mainstream.

REFERENCES

[1] Anders Brodersen, Salvatore Scellato, and Mirjam Wattenhofer. Youtube around the world: Geographic popularity of videos. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, page 241–250, New York, NY, USA, 2012. Association for Computing Machinery.

[2] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 191–198, New York, NY, USA, 2016. Association for Computing Machinery.

[3] Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. Botornot: A system to evaluate social bots. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, page 273–274, Republic and Canton of Geneva, CHE, 2016. International World Wide Web Conferences Steering Committee.

[4] Eslam Hussein, Prerna Juneja, and Tanushree Mitra. Measuring misinformation in video search platforms: An audit study on youtube. *Proc. ACM Hum.-Comput. Interact.*, 4(CSCW1), May 2020.

[5] Jonas Kaiser and Adrian Rauchfleisch. The implications of venturing down the rabbit hole. *Internet Policy Review*, 8(2):1–22, 2019.

[6] Denis Kotkov, Shuaiqiang Wang, and Jari Veijalainen. A survey of serendipity in recommender systems. *Knowledge-Based Systems*, 111:180–192, 2016.

[7] Erwan Le Merrer and Gilles Trédan. The topological face of recommendation. In *International Conference on Complex Networks and their Applications*, pages 897–908. Springer, 2017.

[8] Mark Ledwich and Anna Zaitsev. Algorithmic extremism: Examining youtube's rabbit hole of radicalization. *arXiv preprint arXiv:1912.11211*, 2019.

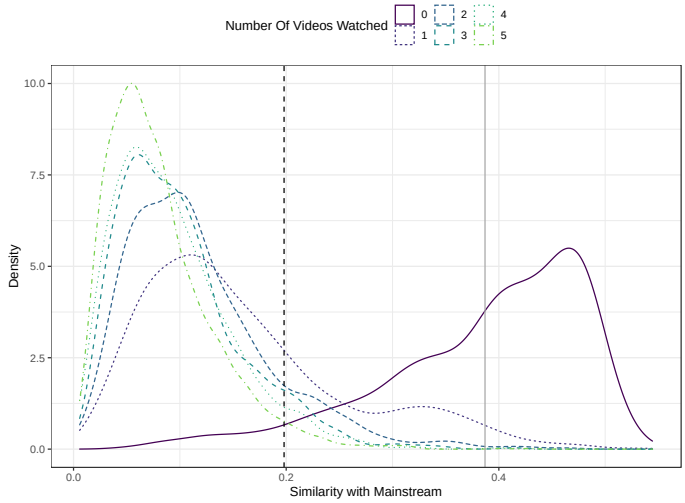

Fig. 7. Distribution of cosine similarity against average mainstream recommendation, as a function of the number of watched videos.

[9] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

[10] Qing Li and Byeong Man Kim. Clustering approach for hybrid recommender system. In *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*, pages 33–38, 2003.

[11] Kai Lukoff, Ulrik Lyngs, Himanshu Zade, J. Vera Liao, James Choi, Kaiyue Fan, Sean A. Munson, and Alexis Hiniker. *How the Design of YouTube Influences User Sense of Agency*. Association for Computing Machinery, New York, NY, USA, 2021.

[12] Saleem Masadeh and Bill Hamilton. The aftermath of the adpocalypse: Systemic bias on youtube. *CHI*, 2020.

[13] Derek O'Callaghan, Derek Greene, Maura Conway, Joe Carthy, and Pádraig Cunningham. Down the (white) rabbit hole: The extreme right and online recommender systems. *Social Science Computer Review*, 33(4):459–478, 2015.

[14] Kostantinos Papadamou, Savvas Zannettou, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Michael Sirivianos. Understanding the incel community on youtube. *CSCW*, 2021.

[15] Eli Pariser. *The filter bubble: What the Internet is hiding from you*. Penguin UK, 2011.

[16] Frank Pasquale. *The black box society*. Harvard University Press, 2015.

[17] Inioluwa Deborah Raji, Andrew Smart, Rebecca N White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. Closing the ai accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 33–44, 2020.

[18] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[19] Manoel Horta Ribeiro, Raphael Ottoni, Robert West, Virgílio A. F. Almeida, and Wagner Meira. Auditing radicalization pathways on youtube. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 131–141, New York, NY, USA, 2020. Association for Computing Machinery.

[20] Camille Roth, Antoine Mazières, and Telmo Menezes. Tubes and bubbles topological confinement of youtube recommendations. *PloS one*, 15(4):e0231703, 2020.

[21] Hemant C Sashittal and Avan R Jassawalla. Brands as personal narratives: learning from user–youtube–brand interactions. *Journal of Brand Management*, 28(6):657–670, 2021.

[22] Christian Stöcker and Mike Preuss. Riding the wave of misclassification: How we end up with extreme youtube content. In *International Conference on Human-Computer Interaction*, pages 359–375. Springer, 2020.

[23] Jonathan Stray, Ivan Vendrov, Jeremy Nixon, Steven Adler, and Dylan Hadfield-Menell. What are you optimizing for? aligning recommender systems with human values. *arXiv preprint arXiv:2107.10939*, 2021.

[24] Adisak Sukul, Baskar Gopalakrishnan, Wallapak Tavanapong, and David A.M. Peterson. Online video ad measurement for political science research. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2132–2140, 2017.

[25] Rashid Tahir, Faizan Ahmed, Hammas Saeed, Shiza Ali, Fareed Zaffar, and Christo Wilson. Bringing the kid back into youtube kids: Detecting inappropriate content on video streaming platforms. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '19, page 464–469, New York, NY, USA, 2019. Association for Computing Machinery.

[26] Matus Tomlein, Branislav Pecher, Jakub Simko, Ivan Srba, Robert Moro, Elena Stefancova, Michal Kompan, Andrea Hrckova, Juraj Podrouzek, and Maria Bielikova. *An Audit of Misinformation Filter Bubbles on YouTube: Bubble Bursting and Recent Behavior Changes*, page 1–11. Association for Computing Machinery, New York, NY, USA, 2021.

[27] Joe Whittaker, Seán Looney, Alastair Reed, and Fabio Votta. Recommender systems and the amplification of extremist content. *Internet Policy Review*, 10(2):1–29, 2021.