How Audit Methods Impact Our Understanding of YouTube's Recommendation Systems

Sarmad Chandio[1], **Muhammad Daniyal Pirwani Dar**[2], **Rishab Nithyanand**[1]

[1]University of Iowa
[2]Stony Brook University
{sarmad-chandio, rishab-nithyanand}@uiowa.edu
mdar@cs.stonybrook.edu

Abstract

Computational audits of social media websites have generated data that forms the basis of our understanding of the problematic behaviors of algorithmic recommendation systems. Focusing on YouTube, this paper demonstrates that conducting audits to make specific inferences about the underlying content recommendation system is more methodologically challenging than one might expect. Obtaining scientifically valid results requires considering many methodological decisions, and each of these decisions incurs costs. For example, should an auditor use logged-in YouTube accounts while gathering recommendations to ensure more accurate inferences from the collected data? We systematically explore the impact of this and many other decisions and make important discoveries about the methodological choices that impact YouTube's recommendations. Taken together, our research suggests auditing configurations that can be used by researchers and auditors to reduce economic and computing costs, without sacrificing inference quality and accuracy.

1 Introduction

The importance of auditing content recommendation systems is increasing. As social media platforms and the algorithms they employ continue to strongly influence our sociopolitical realities, auditing them (accurately) has become increasingly important. After all, effective regulation around online platforms and their algorithms will rely significantly on audits of algorithmic recommendation systems (WH2 2022) and their role in influencing problematic societal behaviors such as political polarization (Barberá 2020), misinformation spread (Hussein, Juneja, and Mitra 2020), and others. For example, focusing on the YouTube platform, researchers have uncovered several concerning aspects of algorithmic recommendations systems, including the propensity to create filter-bubbles (Tomlein et al. 2021), recommend age-inappropriate content (Papadamou et al. 2019), misinformation (Tomlein et al. 2021; Hussein, Juneja, and Mitra 2020), and even extremist content (Ribeiro et al. 2020; Papadamou et al. 2020). However, contradictions are frequently found in these prior research efforts — e.g., researchers have claimed that the YouTube recommendation system exhibits mainstreaming (i.e., promoting popular content over niche content) effects (Ledwich and Zaitsev 2019) as well as the contrasting tendency to promote more niche and extreme content (Ribeiro et al. 2020). Formulating effective regulation and developing a meaningful understanding of the impact of algorithms on society is challenging in these scenarios where contradictory findings are commonplace in algorithm studies. *This work uses YouTube as a case study to show (1) that auditing methods are one source for such contradictory results and (2) approaches for measuring the influence of methods on audit inferences.*

Conceptually, designing a recommendation algorithm audit is simple. Researchers rely on the "sock puppet" audit approach (Sandvig et al. 2014) due to the opacity of the algorithms being audited. The audit can generalize to the following three-step process.

1. Create sock-puppets. Researchers create sock-puppets (i.e., personas) that aim to impersonate real human users. The goal is to use automation tools, typically web crawlers, to provide the underlying recommendation system a set of interactions from which it may learn certain characteristics about the sock puppet. In the context of YouTube, this may involve having the sock-puppet load a set of videos (referred to as the training set) that provide a base from which the recommendation algorithms learn user preferences.

2. Measure the recommendation tree. The sock puppet performs a seed interaction with the algorithm in this step. This interaction generates recommendations that form the first layer of the recommendation tree. Recursively interacting with these recommendations creates a deeper tree of recommendations. Applied to YouTube, this step involves providing the sock-puppet with a seed video from which all recommendations are gathered. This is followed by then loading the videos associated with each of these recommendations themselves to fill the recommendation tree.

3. Hypothesis testing. Finally, researchers test hypotheses and make inferences about the underlying recommendation algorithm. This is done by statistical analysis of the recommendation trees associated with different sock puppets.

In practice, algorithm audits are challenging and force methodological compromises. Although simple at first glance, researchers often overlook the influence of key decisions in each step of the sock puppet auditing process. For example, when conducting crawls to construct sock puppets, researchers are faced with the decisions of what videos to use as part of their sock puppet training set, how many videos to include in this training set, and what video to use as their seed, among others. The uncertainty about the impact that each video might have on the gathered recommendations makes these decisions challenging. Complicating matters, even when rigorous and sound rationale are applied to the above questions, methodological rigor is associated with high dollar and computational costs.

Methodological compromises due to high dollar costs. Online platforms, including YouTube, make it difficult to automate the creation of the required number of accounts for a meaningful audit. Often, web automation tools seeking to create accounts will encounter CAPTCHAs, face outright blocking, or require verifiable phone numbers. Circumventing these challenges can have prohibitively high costs, forcing compromises that can impact the validity of inferences made from the audit. For example, researchers may simply associate each sock-puppet with a unique browser (cookie) and bypass the difficulties (and high costs) associated with obtaining verifiable phone numbers for each sock-puppet. However, such circumvention is done with the assumption that the accuracy of any inferences drawn from the audit are not harmed — i.e., they operate on the assumption that YouTube's algorithms treat logged-in users the same as non-logged-in users with YouTube's cookie in their browser.

Methodological compromises due to high computational costs. YouTube crawlers encounter large numbers of hour-long (or longer) videos, which makes crawling computationally expensive and time consuming. This, combined with the need to gather large amounts of data for statistically sound hypothesis testing, can require 1000's of hours of machine time for a single audit. Thus, an auditor faces many dilemmas: Should one pay the high computational costs associated with watching the entirety of each video? Should one traverse all paths of the recommendation tree to make valid conclusions? Although sampling sections of the tree and not watching videos to completion are tractable alternatives, it is unclear if and how these alternatives will impact the recommendations gathered by the crawl/audit.

Simply put, *there are limited guidelines for sock-puppet-style audits on platforms such as YouTube*. In this paper, we focus on YouTube and fill this gap. We do this by answering the following research questions.

RQ1. What is the relationship between sock-puppet training set, recommendation seed, and recommendation trees? (§3) We begin by studying the impact that the training set and seed have on the recommendation trees they generate. Specifically, we conduct an experiment in which we train four sets of sock-puppets using all combinations of two distinctly different seeds and training sets. We then analyze the recommendation trees they generate to understand how recommendations change with alterations to the seed and training set. Our results show that recommendations have a strong *recency bias*. Consequently, the choice of recommendation seed is significantly more impactful than choice of training set.

RQ2. What is the impact of reducing dollar costs during audits? (§4) We investigate the consequences of one of the most commonly observed cost-saving measures adopted by YouTube auditors — avoiding the use of real YouTube accounts for each sock-puppet and instead relying on browser cookies to leak a sock-puppet's identity to YouTube. We conduct this analysis by comparing the recommendation trees generated by four sets of sock-puppets that reflect commonly observed practices in audit research. These sets of sock-puppets are identical in every way except for their method of maintaining YouTube account 'state.' Our results highlight the feasibility of relying on cheaper alternatives to expensive-to-obtain real YouTube accounts during audits.

RQ3. What is the impact of reducing computational costs during audits? (§5) Finally, we consider the consequences of compromises that are associated with reducing computational costs. We specifically focus on the impact of time spent on each sock-puppet training video and the depth/breadth of recommendation tree exploration. We do so by training sets of sock-puppets that "watch" videos to varying levels of completion and measuring the differences in their gathered recommendation trees. We then study the characteristics of the nodes sampled from all recommendation trees gathered in our study to identify differences in their properties based on their location in the tree. Our results show that several cost-saving mechanisms are possible for auditors, including only watching videos partially.

2 Methodology

We conduct several experiments (*Cf.* Table 1) in which we systematically alter specific audit parameters to uncover their impacts on the recommendation trees they generate. Here, we describe our audit configurations and methods for assessing parameter impact.

2.1 Configuration Parameters

Sock-puppet training sets (T_{niche}, T_{main}). In all experiments, we begin by training our sock-puppets with videos either belonging to a niche (T_{niche}) or a mainstream collection of videos (T_{main}). Although prior work found that 22 videos were enough to personalize the YouTube recommendations (Papadamou et al. 2021), we decided to use 32 videos[1] to ensure the validity of our study.

Niche training set (T_{niche}). The videos in T_{niche} were manually curated to represent unpopular (i.e., lower number of views) and fringe (e.g., conspiracy theories) content. The videos in this set were chosen from fringe subreddits such as *r/climateskeptics* and *r/theworldisflat*, among others. We extracted all the videos from these subreddits and manually selected eight non-mainstream topic-specific videos from each subreddit for addition to T_{niche}. On average, videos in T_{niche} had received 25K views.

Mainstream training set (T_{main}). Each video in T_{main} was manually curated to cover the same topic as its niche counterpart, except that they were sourced from a YouTube search of the topic (e.g., 'flat earth debunked') associated with T_{niche} topics (e.g., 'flat earth'). The eight most popular videos from the search results (based on views) were se-

[1] The full list of videos in each training set is available at: https://osf.io/3j5u8/?view_only=c2e21b99dbc3470e86bc9e904b39e6d3

lected. On average, videos in T_{main} had 5.9M views. This approach of training set construction maintains similarity of topics while offering sharply differing popularity inputs to the recommendation algorithms. This ensures that any effect of training set popularity is measurable.

Recommendation tree seeds (s_{niche} and s_{main}). Seed videos are the starting point from which recommendation trees are gathered (i.e., the root of the recommendation tree). We used one of two seeds (s_{niche} and s_{main}), signifying a niche and mainstream video, which were selected based on the intuition that they would have sharply differing impacts on the recommendation tree. The s_{niche} video used in our experiments was a fringe and unpopular video with 7.1K views and s_{main} was a popular mainstream video with over 3.8M views. The topics of the seed videos were intentionally chosen (1) to not overlap with the any of the videos from the T_{main} or T_{niche} so that effects from the training sets could be distinguished from those of the seed, and (2) to not overlap with each other, on topic or popularity characteristics, to maximize any measurable differences between their recommendation trees. Observing an absence of differences in recommendation trees generated from s_{niche} and s_{main} would indicate that the seed has a marginal influence on the observed recommendations.

Account status parameters (A_{login}, A_{cookies}, and A_{clear}). To improve our understanding about whether the recommendation algorithm works differently when audits operate under different YouTube account configurations, we gather recommendation trees using three different types of account configurations. (1) A_{login} represents audits in which crawlers are logged into freshly created YouTube accounts before training and recommendation gathering begins. This is representative of the ideal case where each sock-puppet has its own fresh YouTube account. (2) A_{cookies} represents audits where crawlers are not logged-in but maintain YouTube's cookies in their browser throughout the crawl. This is representative of the most common crawls observed in audit literature. (3) A_{clear} represents audits where crawlers conduct crawls while logged-in and clear their watch history before the same account is used for another crawl. This approach is used to allow account reuse by different sock-puppets.

Watch times (W_{100pc}, W_{50pc}, W_{25pc}, W_{10pc}). Training sock-puppets can be computationally expensive owing to the long lengths of videos typically contained within the training sets. To understand whether videos in the training set need to be watched to completion, we gather recommendation trees from four crawlers all configured identically except that they watch each of the training videos to different levels of completion before moving on to the next video. W_{100pc}, W_{50pc}, W_{25pc}, and W_{10pc} watch videos to 100%, 50%, 25%, and 10% of completion, respectively.

Interactions (I_{get}, I_{click}). Programming crawlers to perform actual clicks on hyperlinks is a challenging task due to difficulties with reliability. A commonly used alternative is to instead obtain links by parsing the DOM and having the browser load the link of interest. Unfortunately, the absence of actual clicks is also a signature used by common bot-detection tools and may result in server-side differential treatment (Khattak et al. 2016; Ahmad et al. 2020; Jueckstock et al. 2021). We conduct an experiment to understand whether clicking on recommended videos impacts subsequent recommendations. I_{click} represents an audit in which each crawler actually performs a mouse click on videos to load them during the recommendation tree crawl. I_{get} represents an audit in which each crawler simply obtains the video's URL from the DOM and instructs the browser to load that URL.

Question	Parameter	Configurations	#trees	#videos
RQ1	Training set	T_{main}, T_{niche}	16	32K
	Seed video	s_{main}, s_{niche}	16	32K
RQ2	Accounts	A_{login}, A_{cookies}	8	14K
		A_{login}, A_{clear}	8	13K
RQ3	Watch Time	W_{100pc}, W_{50pc}	8	15K
		W_{50pc}, W_{25pc}	8	15K
		W_{25pc}, W_{10pc}	8	15K
	Interaction	I_{get}, I_{click}	8	16K
	Breadth	P_{left}, P_{right}	all*	69K
	Depth	D_{top}, D_{bottom}	all*	35K

*Data from all trees were used in analysis for these parameters.

Table 1: 'Parameters' indicate the values we modified in each experiment. 'Configuration' indicate the values assigned to the parameter in the experiment. *Cf.* §2.1 for descriptions of each parameter and their respective configurations. The '# trees' column indicates the total number of recommendation trees gathered for analysis and the '# videos' column indicates the total number of recommendations observed in these trees.

Breadth of exploration (P_{left}, P_{right}). YouTube's recommendations are dynamically loaded and recommendation options often continue to appear while a user scrolls down the page. This increases the width of the recommendation tree at each level. In our pilot tests, we observed that the minimum number of recommendations was at least 40 for each video (and much higher in many cases). We conduct analyses on the videos that appear at the top of the recommendation list during a recommendation tree crawl (i.e., the left-most path in the tree denoted by P_{left}) and those that appear at the bottom of the recommendation list (i.e., the right-most path in the tree denoted by P_{right}).

Depth of exploration (D_{top}, D_{bottom}). Finally, we consider the importance of performing deep crawls on measured characteristics of the recommendation tree. We do this by analyzing the characteristics of all videos observed after just loading the seed video (i.e., the 1^{st} level in the recommendation tree denoted by D_{top}) and comparing them with the characteristics of all videos observed at the 10^{th} level of the tree (i.e., the bottom of *our* gathered recommendation trees denoted by D_{bottom}).

2.2 Data Gathering

Minimizing the influence of latent confounding variables. Recommendation trees are influenced by a large number of variables, some in researchers' control (e.g., our configuration parameters) and others not. In our study, we make a best-effort attempt to minimize these latent effects with the following approaches.

Accounting for updates to the search index. Due to large amounts of new content being created on YouTube, there are continuous changes to the search index and recommendation candidate lists. Therefore, two crawls gathering recommendations at time periods that are far spaced apart, may not be comparable due to vastly different recommendation possibilities. We mitigate such impacts by synchronizing the crawls conducted in each experiment such that for every crawler using one configuration to gather a recommendation tree, there is another synchronized crawler using the alternate configuration to gather the comparison recommendation tree. This synchronization is done at the node level — i.e., we ensure that each tree arrives at the exact same node position in its respective recommendation tree within, at most, a few seconds of its counterpart. Therefore trees gathered using alternate configurations of the same parameter are comparable.

Accounting for distributed infrastructure and effects of geolocation. As shown in prior work (Hannak et al. 2013), web servers may be distributed across a wide region and servers in different locations or data centers may have inconsistencies in their search indices or perform geo-specific recommendations. To mitigate these effects on our gathered trees, we conduct all our data gathering experiments from the same location and use a static DNS entry for YouTube which ensures that all our content requests and interactions with the platform are served by web servers, at the very least, in the same region.

Accounting for A/B testing. Platforms have been known to conduct A-B testing on their users while testing new features or algorithm updates (Facebook 2022). We make a best-effort attempt to mitigate the effects of such testing by gathering data from *at least eight identical and synchronized crawls for each parameter tested in our study*.

Collecting recommendation trees. Once a sock-puppet has been trained and has a seed video, we begin exploration of the recommendation tree. Unfortunately, complete exploration of a recommendation tree is infeasible due to the need for one sock-puppet for each configuration being tested for each tree being gathered for each path being traversed. This is necessary due to the fact that prior watched videos will impact future recommendations and therefore a sock-puppet can only perform one-way (downward) traversals of the recommendation tree. Further, we are collecting at least 40 recommendations for each video. Therefore, a recommendation tree of depth n will have at least 40^n paths from root to leaf node each needing a unique sock-puppet. In our traversals of the tree, we explored five unique paths — the left-most path (comprised of the first recommendation at each node), the right-most path (comprised of the last recommendation at each node), and three pre-selected paths from the middle (sampled with zipfian weights to account for a preference for videos higher in the recommendation list). We explore each of these paths simultaneously, using a unique but identically trained, configured, and seeded sock-puppet dedicated to each, to a depth of ten and record all recommendations along the way. We stitch these paths and observations together to obtain a subset of the complete recommendation tree upon which our analysis is conducted. We gather at least four such trees for each parameter configuration while ensuring synchronization with alternately configured audits. An example of such a tree is shown in Figure 1.

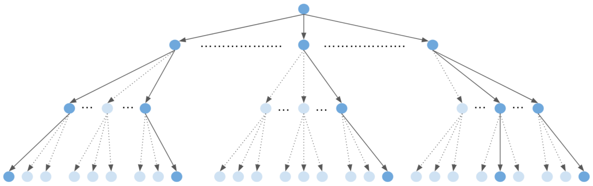

Figure 1: Each path (flat arrows) in this figure represents the set of videos that form the sock-puppets, nodes represent videos, and directed edges between any two nodes ($parent$, $child$) indicate that $child$ was recommended after direct interaction with the $parent$. The root of this tree represents the seed video used to generate the first set of recommendations.

2.3 Recommendation Tree Characteristics

In our analysis, we focus on studying the influence of the aforementioned crawl configurations (§2.1) on the popularity, channel diversity, and topics of videos observed their corresponding recommendation trees. We select these characteristics since platform audits often focus on them (or their variations) to identify echo-chamber, rabbit-holing, or mainstreaming effects caused by recommendation algorithms.

Popularity of recommendations. *Popularity of recommended content, measured using video views as a proxy, can capture the algorithm's tendency to recommend niche or mainstream content.* We record the distribution of views observed in recommended videos at each node. A recommendation tree largely containing videos with low popularity at each node suggests the tendency to recommend niche content for the associated sock-puppet configuration. Conversely, a tree largely containing videos with high popularity at each node suggests the tendency to recommend mainstream content for the associated sock-puppet configuration. Significant differences in the within- and across-group differences between the trees generated by two configurations would suggest that one of the two configurations tends to more mainstream (popular) recommendations than the other.

Channel diversity of recommendations. Each video about a topic reflects the unique perspective of the channel that uploaded the video and the community that consumes it. *Therefore, we use the diversity of channels in the trees as an approximation for the range of perspectives provided by the recommended content.* We take the position that even when two channels discuss the same topic, they frequently produce different perspectives due to differences in their creative teams and communities. Therefore, a recommendation tree with a high entropy of recommended content at each node indicates high recommendation diversity and suggests at the absence of a rabbit-holing effect. We measure the channel diversity by recording the entropy of channels observed in the recommended content at each node. Significant differences in the within- and across-group differences between the trees generated by two configurations would suggest that one of the two configurations tends to show less diverse recommendations than the other.

Semantic similarity of recommendations. We extract the titles and descriptions associated with each video observed in our tree. *We combine these processed texts for all videos observed at each node in a tree and use it as a representation of the content semantics observed in the recommendations at that node.* To finalize our approach, we conducted the following pilot study: two very similar nodes (of forty videos each) and two dissimilar nodes (forty videos each) were extracted from our dataset whose similarity between texts was manually determined by the researchers based on the titles and descriptions [2]. Different text representations, including semantic (LSI (Rosario 2000) and SpaCy's docism (SpaCy 2022)), lexicographic (LDA (Blei, Ng, and Jordan 2003)), and transformer-based (sentence-BERT (Reimers and Gurevych 2019)) were used to calculate the similarity scores. Two experts rated the similarity of the texts. The inter-rater agreement was the highest for the sentence-BERT similarities. Significant within- and across-group differences between trees generated by two configurations suggest that one of the two configurations results in measurably different recommended topics.

Selecting and interpreting metrics. We selected the above three metrics because prior audits have largely focused on evaluating their more complicated variants and functions. For example, Hussein et al. focus on identifying tendencies to recommend misinformative content with specific stances, which we consider as a variant of the channel diversity and topic-related metrics. During interpretation of our results, we focus on holistically considering differences in all three metrics to understand the impact of parameters on recommendations. This allows us to make nuanced determinations of the underlying algorithms. For example, observing significant differences in the 'channel diversity' metric but not in the 'semantic similarity' metric suggests a different perspective on the rabbit-holing tendencies of the recommendation algorithms. Further, observing significant difference in all three metrics allows us to make stronger claims regarding the influence of the corresponding parameter configuration.

2.4 Comparison of Audit Configurations

Each of our experiments result in two sets of recommendation trees — one set for each audit parameter configuration being tested (e.g., A_{login} and A_{cookies}). Trees in each set are gathered in synchronization with each other. Given these sets of recommendation trees, we compute the *across-group* (e.g., between A_{login} and its synchronized A_{cookies} tree) and *within-group* (e.g., between two synchronized A_{login} (or, A_{cookies}) trees) differences along the three dimensions described in §2.3. We describe this process below.

Recording characteristics of a recommendation tree node. Let n_{ij} denote a traversed node (i.e., viewed video) located on path P_i and at depth j in a recommendation tree and r_{ijk} denote the k^{th} recommendation observed at n_{ij}. At each node n_{ij} we record: (1) a popularity scalar value $pop(n_{ij}) = \mu(views(r_{ij,1}) \dots views(r_{ij,40}))$ representing the view counts of all observed recommended videos at this node; (2) a channel entropy scalar value $div(n_{ij}) = entropy(channel(r_{ij,1}), \dots, channel(r_{ij,40}))$ representing the diversity of channels in the recommended videos at this node; and (3) a document vector $doc(n_{ij}) = docvec(desc(r_{ij,1}), \dots, desc(r_{ij,40}))$ which represents the document vector associated with the video descriptions obtained from all recommended videos at this node.

Comparing characteristics of recommendation trees. Given two recommendation trees T and T', we compute the differences in characteristics in a node position-dependant manner — i.e., we compute differences in the popularity vector, channel entropy, and document vector for each node position in T and T'. These differences are computed as follows:

$$\delta_{pop}(T, T') = mean([\forall i, \forall j : pop(n_{ij}) - pop(n'_{ij})])$$

$$\delta_{div}(T, T') = mean([\forall i, \forall j : div(n_{ij}) - div(n'_{ij})])$$

$$\delta_{sem}(T, T') = mean([\forall i, \forall j : sim(doc(n_{ij}), doc(n'_{ij}))])$$

These values effectively capture the mean node-to-node differences between T and T'. This node-to-node comparison is possible because all trees gathered in our study traversed the same set of paths in the recommendation tree. Maintaining this node position dependence in tree comparisons is important because it handles differences in characteristics that might arise from the position of a node in the recommendation tree. For example, comparing the first recommendation at depth=1 from T with the 40^{th} recommendation at depth=10 from T' could result in misattributing differences in tree characteristics that arise from changes in recommendation ranks to the impact of an audit configuration change.

Computing within- and across-group differences. Given two auditing configurations $\mathcal{C}$ and $\mathcal{C}'$ which generate the sets of trees $\mathcal{T}$ and $\mathcal{T}'$, respectively, we compute: (1) the *within-group differences* as the distribution of differences in characteristics observed between trees within $\mathcal{T}$ and $\mathcal{T}'$; and (2) the *across-group differences* as the distribution of differences observed between trees across $\mathcal{T}$ and $\mathcal{T}'$. These are denoted by:

$$\Delta_x^{within}(\mathcal{T}) = [\forall (T_i, T_j) \in (\mathcal{T} \times \mathcal{T}) : \delta_x(T_i, T_j)]$$

$$\Delta_x^{across}(\mathcal{T}, \mathcal{T}') = [\forall (T_i, T_j) \in (\mathcal{T} \times \mathcal{T}') : \delta_x(T_i, T_j)]$$

$$\forall x \in \{pop, div, sem\}$$

The within-group differences, computed over all trees generated with identical audit configurations, allow us to establish a *baseline* of characteristic variations caused by factors outside the control of the auditor (e.g., probabilistic recommendation algorithm, A/B testing, etc.). The across-group differences showcase the differences caused by the change in audit configuration *and* external factors.

Quantifying the impact of audit parameter configurations. Given distributions Δ_x^{within} and Δ_x^{across} associated

[2] We use a pre-trained MPNet model (Song et al. 2020) to compute our sentence embeddings.

Parameters			**Video Popularity**			**Channel Diversity**		**Content Semantics**	
			(Views in millions)			**(Entropy in bits)**		**(Similarity score)**	
Fixed	Varied	μ_{views}	Effect (95% CI)	μ_{effect}	μ_{entropy}	Effect (95% CI)	μ_{effect}	Effect(95% CI)	μ_{effect}
s_{main}	T_{main} T_{niche}	7.15 4.94	**[0.34, 1.33]**	0.84	3.63 3.49	[-0.16, 0.17]	0.00	[-0.06, 0.01]	-0.03
s_{niche}	T_{main} T_{niche}	4.32 1.80	**[1.46, 2.19]**	1.82	3.38 3.26	[-0.27, 0.19]	-0.04	**[-0.10, -0.03]**	-0.06
T_{main}	s_{main} s_{niche}	10.71 7.78	**[0.73, 2.31]**	1.51	3.17 2.97	[-0.14, 0.14]	0.00	**[-0.12, -0.05]**	-0.08
T_{niche}	s_{main} s_{niche}	4.93 1.72	**[2.68, 3.05]**	2.87	4.02 3.44	**[0.12, 0.45]**	0.28	**[-0.10, -0.03]**	-0.06

Table 2: Impact of changes caused by varying training sets (top 2 rows) and seeds (bottom 2 rows). Columns represent the mean node values observed in each group for a particular characteristic, the 95% confidence interval for the measured effect sizes (i.e., difference between within- and across-group differences; *Cf.* §2.4), and the mean effect size. Values in bold indicate a statistically significant effect size at the corresponding confidence level.

with configurations ($\mathcal{C}$, $\mathcal{C}'$), we use bootstrapping with 1M samples (DiCiccio and Efron 1996; Efron 1987) to create 95% confidence intervals around the mean within- and across-group differences. We also use these bootstrapped samples to compute 95% confidence intervals around the effect size — i.e., *the difference between the within- and across-group differences bootstrap samples*. Let $[CI_{lower}, CI_{upper}]$ be the N% confidence interval for the effect size. The effect is statistically significant at this confidence level if and only if $(CI_{lower} \leq CI_{upper} < 0)$ or $(CI_{upper} \geq CI_{lower} > 0)$ — i.e., *iff N% of the bootstrapped samples have observed effect sizes of the same polarity*. In our work, we report the 95% confidence interval for effect sizes. We also report the average effect size as the mean of all effect sizes observed in the bootstrap samples.

3 Training Sets and Seeds

Experiment setup. Our goal is to measure the impact of training sets and seeds on the characteristics of recommendation trees generated by an audit. To accomplish this, we gathered 32 recommendation trees from four different audit configurations: eight trees each from an audit using T_{main} and s_{main}, T_{main} and s_{niche}, T_{niche} and s_{main}, and T_{niche} and s_{niche}. We split each of these into two sets of four and refer to them as ($\mathcal{T}_{\text{main,main}}$, $\mathcal{T}'_{\text{main,main}}$), ($\mathcal{T}_{\text{main,niche}}$,$\mathcal{T}'_{\text{main,niche}}$), ($\mathcal{T}_{\text{niche,main}}$,$\mathcal{T}'_{\text{niche,main}}$), and ($\mathcal{T}_{\text{niche,niche}}$,$\mathcal{T}'_{\text{niche,niche}}$) respectively. These trees were gathered in synchrony (*Cf.* §2.2) in order to facilitate accurate within- and across-group comparisons (*Cf.* §2.4). By splitting each of our sets of eight trees into two sets of four, we avoid reusing trees for testing multiple hypotheses.

Measuring impact of a training set change. To uncover the impact of the training set used in an audit on the characteristics of recommendation trees, we compute the means, 95% confidence interval associated with the within-group differences, across-group differences, and effect sizes (*Cf.* §2.4) obtained from two analyses: (1) comparing $\mathcal{T}_{\text{main,main}}$ with $\mathcal{T}_{\text{niche,main}}$ — i.e., using the same mainstream seed while varying the training set; and (2) comparing $\mathcal{T}_{\text{main,niche}}$ with $\mathcal{T}_{\text{niche,niche}}$ — i.e., using the same niche seed while varying the training set.

Measuring impact of a seed change. We repeat our method for the following analyses: (1) comparing $\mathcal{T}'_{\text{main,main}}$ with $\mathcal{T}'_{\text{main,niche}}$ — i.e., varying the seed while using a mainstream training set focused on controversial topics; and (2) comparing $\mathcal{T}'_{\text{niche,main}}$ with $\mathcal{T}'_{\text{niche,niche}}$ — i.e., varying the seed while maintaining a fringe and controversial training set.

Results. Our results are summarized in Table 2. In general, we find that altering the characteristics of the training set or the seed *always* impacts the popularity of the videos observed in an audit. This, however, is not the case for the channel diversity and semantics. More specifically, our analysis yields the following insights.

There appears strong evidence of a 'recency bias' in recommendations. Paying attention to the bottom two rows of Table 2, we see that the effects of altering the seed from a niche video to a mainstream video are nearly always statistically significant and of high magnitude, with only one exception when channel diversity is recorded using T_{main} for training. The (significant) effects on the popularity and entropy of recommended videos are also higher than the effects observed on alterations of the training set (top two rows). The most notable effects of altering seeds are in the 'popularity' dimension where the mean effect of switching a seed video from niche to mainstream results in video recommendations that, on average, have 1.51M and 2.87M more views when trained with T_{main} and T_{niche}, respectively. Surprisingly, the highest effect size in the semantics of recommended videos also occur when only seeds are altered from mainstream to niche and training sets are kept the same. Focusing on the bottom two entries of Table 2, we observe that on average, recommended videos are 8% and 6% less semantically similar just by seed alteration when trained with T_{main} and T_{niche} respectively. A deeper look into the tree level topics (focusing on the top 12) reveals striking differences. With mainstream seed, recommendations included topics like *scandal* involving Will Smith, J. Depp and Amber Heard. Health and medicine related *science* topics were also prevalent. In the domain of *religion*, terms such as "bible" and "jesus" were dominant. Conversely, the niche seed resulted in *news* being the dominant topic, with Fox News contributing the most. This was followed by *political activism*, highlighted by mentions of J.B Peterson, Chomsky, and Mearsheimer. In the *religious* cluster, terms like "Aquinos", "catholic", "christian", and "white-america" were most visible. What's intriguing is that the mainstream seed did make occasional recommendations aligned with dominant themes from the niche seed, such as mentions of J.B Peterson and Fox News. However, the opposite was not observed: the niche seed did not suggest any predominant mainstream topics. This suggests that, independently of the training set used, the choice of seed can drastically alter the characteristics of a recommendation tree and the audit inferences. Extrapolating this finding suggests that the most recent video will have an outsized impact on future recommendations as shown in Figure 2.

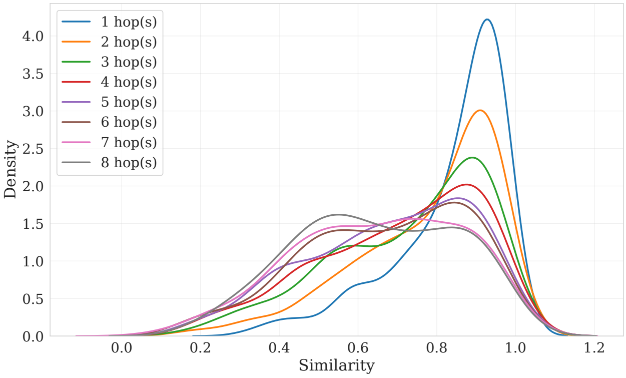

Figure 2: The figure displays the similarity between nodes that are separated by 'n' steps (or 'n-hops') within the same path and tree. It includes data from all the configurations (except [P_{left}, P_{right}] and [D_{top}, D_{bottom}]). A noticeable pattern emerges: nodes further down the same path are less (semantically) similar.

Channel diversity is not always dependent on the training set and seed. Our analysis shows that the channel diversity is largely unaffected by the choice of training set and seed. Only one exception occurs: when seeds are altered for a T_{niche} training set audit (*Cf.* row four in Table 2). Here we see the effect of switching from s_{main} to s_{niche} reduces the channel diversity by an average of 0.28 (entropy in bits) at each node. While it appears that this finding lends credence to the claims of the algorithms' rabbit-holing tendencies, it is important to note that this decrease only appears when the audit has interacted with fringe content (in the training set and the seed). Given that the effect disappears when any other interaction occurs, this finding could be explained by the small number of creators addressing the topic of the niche content.

Takeaways. Assessed together, these results put a different perspective on YouTube's recommendation system and the audits that study it. Not only do researchers need to pay particular attention to training and seeding, but also must understand that their measurements of recommended videos are heavily dependent on the *most recent* nodes already traversed by their sock-puppets. Specifically, it appears that the recency bias can lead to a single video overwhelming the effects of a large number of prior videos — thus impacting the final inferences from the audit. Generally, we recommend that audit inferences (e.g., presence of a mainstreaming effect) are conditioned: (1) on the specific characteristics of the training set and seed; and (2) on the specific strategies used to select nodes from a recommendation tree.

4 Dollar-Cost Saving Configurations

Experiment setup. In this section, we focus on understanding the impact on recommendation trees generated by commonly used sock-puppet account management strategies (e.g., login vs cookies, etc.)

Measuring the effectiveness of cookie-based sock puppets. To find out the differences in cookie-based sock puppets against real accounts, we gathered four recommendation trees for $\mathcal{T}_{\text{full}}$ and $\mathcal{T}_{\text{cookies}}$ each. All the parametric configurations for these two sets were kept identical except $\mathcal{T}_{\text{full}}$ was using a logged-in profile while $\mathcal{T}_{\text{cookies}}$ was only maintaining YouTube cookies. Both $\mathcal{T}_{\text{full}}$ and $\mathcal{T}_{\text{cookies}}$ used the (T_{main}, s_{main}) training set and seed.

Measuring the effectiveness of clearing account history. To verify whether clearing account history purges the watch history effect (i.e even after deleting watch history, user keeps getting similar recommendations), we collected four recommendation trees for $\mathcal{T}'_{\text{full}}$ and $\mathcal{T}_{\text{clear}}$ each. Both $\mathcal{T}'_{\text{full}}$ and $\mathcal{T}_{\text{clear}}$ use logged-in profiles (T_{main}, s_{main}) training set and seed. However, before collecting recommendations, watch history of $\mathcal{T}_{\text{clear}}$ was deleted.

Results. The results are summarized in Table 3. Our analysis yielded two conclusive results.

Audits do not need fresh accounts for each sock-puppet. First, focusing on the impact of changing between a sock-puppet with a logged-in YouTube account ($\mathcal{T}_{\text{login}}$) and one which only maintains its browser cookies ($\mathcal{T}_{\text{cookies}}$), we found that there were no significant differences in any measured characteristics of their recommendations. Which means that the content personalization experience for both a logged-in account and a browser instance with cookies is the same. Based on the marginal effect sizes across all three dimensions of row one in Table 3, we hypothesize that the objective function of the recommendation system for cookies and free accounts is likely aligned. This may be due to neither configuration involving monetary payment, and revenue being primarily generated through ads. This presents significant cost-saving opportunities that arise from being able to associate a sock-puppet with a browser instance rather than having to navigate the barriers associated with automating account creation and phone number verification.

The potential for account reuse by clearing history. There is a significant difference in popularity and content seman-

Parameters		**Video Popularity** **(Views in millions)**			**Channel Diversity** **(Entropy in bits)**		**Content Semantics** **(Similarity score)**	
	μ_{views}	Effect (95% CI)	μ_{effect}	μ_{entropy}	Effect (95% CI)	μ_{effect}	Effect (95% CI)	μ_{effect}
A_{login} A_{cookies}	9.20 7.72	[-0.90, 0.99]	0.05	3.36 3.57	[-0.11, 0.31]	-0.01	[-0.06, 0.03]	-0.01
A_{clear} A_{login}	12.34 8.47	**[1.82, 3.46]**	2.65	3.55 2.86	[-0.03, 0.53]	0.26	**[-0.12, -0.05]**	-0.08

Table 3: Impact of changes caused by varying login status (row 1) and purging watch history (row 2). Columns represent the mean node values observed in each group for a particular characteristic, the 95% confidence interval for the measured effect sizes (i.e., difference between within- and across-group differences; *Cf.* §2.4), and the mean effect size. Values in bold indicate a statistically significant effect size at the corresponding confidence level.

tics for $\mathcal{T}_{\text{full}}$ sock-puppets when compared with identically configured and synchronized $\mathcal{T}_{\text{clear}}$ sock-puppets, suggesting that, by clearing history $\mathcal{T}_{\text{clear}}$ has reset the popularity-context and topic-context (picked up during training phase) which $\mathcal{T}_{\text{full}}$ still maintains. Simply put, by clearing account history, one might be able to reuse an account for a large-scale study — particularly where the popularity and content semantics are being measured (e.g., in audits quantifying mainstreaming and rabbit-holing effects). However, we do not make the claim that clearing watch history is equivalent to creating a fresh account (a fresh account would mean Google doesn't have any data stored for the profile at the back-end, which we did not check for).

Takeaways. These findings present an opportunity for auditors to save huge dollar-costs involved in account creation and curation. We have shown that a browser that maintains YouTube cookies is as good as a YouTube account. Furthermore, account re-use (after clearing history) is a viable option for auditors studying the platform for its popularity and content semantics.

5 Computational Compromises

Experiment setup. In this section, we analyze the impact of three compromises that may be made to save computational resources: (1) watching only a pre-determined fraction of each video in the recommendation tree; (2) using the `driver.get(URL)` method of selenium rather than automating user clicks on recommended videos through `ActionChains(driver)`; and (3) performing low-depth and narrow-breadth audits.

Measuring impact of video watch times. To answer whether audits need to 'watch' videos to completion, we gathered and analyzed four recommendation trees in which the audit 'watched' all videos to completion ($\mathcal{T}_{\text{w=100}}$), eight trees in which the audit only 'watched' videos to 50% of their duration ($\mathcal{T}_{\text{w=50}}$, $\mathcal{T}'_{\text{w=50}}$), and four trees in which the audit only 'watched' videos to 25% of their duration ($\mathcal{T}_{\text{w=25}}$). Both sets of audits used the (T_{main}, s_{main}) training set and seed.

Measuring impact of interaction mechanics. We gathered four recommendation trees where the audit actually located and clicked the recommendations video links ($\mathcal{T}_{\text{click}}$) and four trees where the audit simply identified the URL of the recommended videos and fetched the video with a `driver.get(URL)` command ($\mathcal{T}_{\text{get}}$). Both sets of audits used the (T_{main}, s_{main}) training set and seed.

Measuring the impact of crawl-breadth and -depth. We analyzed the characteristics of the leftmost and rightmost paths of all 96 recommendation trees gathered in this study ($\mathcal{T}_{\text{left}}$ and $\mathcal{T}_{\text{right}}$). These correspond to the paths obtained from only clicking the top and bottom recommendation at each video, respectively. We also analyzed the characteristics of the recommendations observed at depth 1 and 10 for all 96 trees obtained in this study ($\mathcal{T}_{\text{top}}$ and $\mathcal{T}_{\text{bottom}}$).

Results. Our results are shown in Table 4. Notably, besides configurations with varying crawl depth, none of our changes yielded statistically significant differences in their measured recommendation characteristics. This has several key implications for auditors.

Videos do not need to be watched to completion. In all our audit configurations that varied video watch time fractions, there was no statistical relationship between change in the characteristics of recommended videos and the audit's configured watch fraction. This is a surprising finding that suggests even watching 10% of a video impacts the subsequent recommendations to no different extent as watching 100%. Upon further investigation, we discovered evidence showing that YouTube only requires a watch time of 30 seconds (sometimes even 10 seconds) or intentional initiation of watching a video for a 'view' to be registered (Ram and Davim 2018; Parsons 2017; Funk 2020). Based on these speculations, we hypothesize that these 'view' metrics are also used to determine whether a video should impact subsequent recommendations. Since, we were interacting with all the videos by turning off auto play, skipping ads, and pausing as part of synchronous crawls, it might have depicted user-intentionality. It's also plausible that many of the videos had lengths greater than 30 seconds, even if we watched 10% of them. This finding that videos do not need to be watched to any specific fraction of completion presents a promising (accuracy-independent) computational cost-saving avenue for future auditors.

It is unnecessary to automate clicks on recommended videos. Our analysis showed no statistically significant differences between any recommendation tree characteristics observed in $\mathcal{T}_{\text{get}}$ and $\mathcal{T}_{\text{click}}$. This suggests that using browser automation tools (e.g., Selenium webdriver's action chains) to ex-

Parameters	**Video Popularity (Views in millions)**			**Channel Diversity (Entropy in bits)**			**Content Semantics (Similarity score)**	
	μ_{views}	Effect (95% CI)	μ_{effect}	μ_{entropy}	Effect (95% CI)	μ_{effect}	Effect (95% CI)	μ_{effect}
W_{100pc}	7.69	[-0.86, 0.28]	-0.29	3.74	[-0.23, 0.22]	0.00	[-0.05, 0.00]	-0.02
W_{50pc}	7.84			3.51				
W_{50pc}	12.13	[-3.52, 1.53]	-1.03	3.21	[-0.41, 0.15]	-0.13	[-0.05, 0.05]	0.00
W_{25pc}	9.55			3.60				
W_{25pc}	14.11	[-2.10, 0.43]	-0.85	3.61	[-0.56, 0.12]	-0.22	[-0.03, 0.06]	0.01
W_{10pc}	13.59			3.47				
I_{click}	7.62	[-0.59, 0.64]	0.02	3.79	[-0.20, 0.11]	-0.04	[-0.03, 0.05]	0.01
I_{get}	6.93			3.88				
P_{left}	8.33	[-0.65, 0.98]	0.16	3.72	[-0.03, 0.20]	0.08	**[-0.08, -0.03]**	-0.06
P_{right}	7.47			3.33				
D_{top}	13.73	**[5.04, 6.67]**	5.86	4.59	**[1.05, 1.24]**	1.14	**[-0.12, -0.06]**	-0.09
D_{bottom}	5.97			3.12				

Table 4: Impact of changes caused by varying video watch times (rows 1-3), interaction mechanisms (row 4), recommendation selection strategy (row 5), and crawl depth (last row). Columns represent the mean node values observed in each group for a particular characteristic, the 95% confidence interval for the measured effect sizes, and the mean effect size. Values in bold indicate a statistically significant effect size at the corresponding confidence level.

plicitly click on video links is unnecessary. Without sacrificing on accuracy of audit inferences, this allows auditors to replace a computationally expensive, high programmer overhead, and unreliable approach to navigate to subsequent recommendations with the simple and reliable approach of programming browsers to fetch specific URLs in the DOM.

Crawl depth impacts recommendation characteristics. Our analysis on the impact of crawl-depth yields statistically significant results for all recommendation tree characteristics. Specifically, we notice that nodes at the top of the recommendation tree generally appear to be significantly more popular, diverse, and less semantically similar to recommendations at the bottom of the tree. This finding once again showcases the possibility of a strong recency bias that impacts recommendations. Interestingly, we do not see statistically significant differences between the highest- and lowest-recommended videos — suggesting that auditors need to pay specific attention to the depth of their crawls.

Takeaways. Our analysis yields two significant computational cost-savings for researchers. Specifically, finding that videos do not need to be watched to completion and that clicking on videos causes no different outcomes than simply 'getting' the URL of the corresponding video reduces the computational and engineering overhead associated with an audit. In addition, our work highlights that different depths of a recommendation tree could result in different recommendation characteristics. To account for these effects, it is important that any inferences from an audit are conditioned on the depth of the trees that were used.

6 Related Work

Algorithmic audits. Sock puppets and other controlled approaches (Metaxa et al. 2021) are a way to check if the algorithm aligns with the expected behavior. To better understand the impact of the algorithm on the users, ecological studies, e.g extension-based based methods (Hosseinmardi et al. 2021; Chen et al. 2023) are more useful, as they are more representative of the real user behavior. There has been a lot of work discussing the importance of algorithmic audits and devising general guidelines on conducting them (Sandvig et al. 2014; Metaxa et al. 2021; Goodman and Trehu 2023). These studies have a conceptual approach to auditing, and primarily emphasize the theoretical importance of various methodologies. While our work draws inspiration from these studies, it differs from them by taking an empirical approach. We demonstrate that even when focusing on a single methodology, there are intricate details that affect the reproducibility of the study. Consequently, our research is dedicated to formulating guidelines for sock-puppet-style audits on platforms similar to YouTube.

Audits of YouTube's recommendation system. This paper was inspired by a recent influx of YouTube audit research which often showed contrary results. For instance, Lutz et al. (2021) provided evidence of the absence of a rabbit-holing effect while demonstrating a mainstreaming effect for a variety of political ideologies. Other work (Ledwich and Zaitsev 2019; Munger and Phillips 2022; Hosseinmardi et al. 2021; Makhortykh and Urman 2020) has also challenged the notion of rabbit-holing on YouTube and shown evidence of recommendations swaying users towards mainstream and neutral content. Contrary to these findings, Haroon et al. (2022) provided evidence that YouTube pushes users towards increasingly biased and radical political content on 'up-next' and homepage recommendations. These findings are complementary to another body of work (Bryant 2020; Ribeiro et al. 2020; Tomlein et al. 2021; Papadamou et al. 2019, 2021) which has argued that YouTube recommendations have promoted polarization in the political, scientific, and health-related domains. While differences may arise due to the qualitative or quantitative nature of a study, it is concerning to find contradictions when the two methodologies are comparable. Comparing the recent works of Ibrahim et al. (2023) and Haroon et al. (2022) which conclude the YouTube algorithm is right and left leaning respectively, we find evidence of potentially problematic differences in their methods including differences in the way they maintained account state, watch fractions, and even seed video selection. As shown in our analysis, each of these can have significant influences on the characteristics of recommended content. Unlike these previous efforts, our goal is not to support or undermine specific theories about YouTube's tendency to impact polarization. Rather, we aim to uncover the possible reasons for these differences and provide guidelines to avoid such confusion and contradictions within the auditing community. More recently, Ribeiro et al. (2023) presents evidence for de-amplification of niche content for a utility based model. In contrast, a blind model (a simulation which only follows the up-next path), always got increasingly exposed to niche content. In a study focusing on YouTube's demonetization algorithm, Dunna et al. (2022) found evidence that the recommendation and demonetization algorithms were linked. There are also numerous publications from Google describing the recommendation algorithm used for YouTube. These have suggested the use of user profiles, watch histories, video watch times, and click-through rates as features in their content ranking algorithm (Zhao et al. 2019; Tang et al. 2019; Fu et al. 2016; Covington, Adams, and Sargin 2016; Zhao et al. 2015). These descriptions informed our choice of audit parameters.

Improving the reliability of crawler-based research. There have been similar efforts to ours in the Internet measurement community. These have largely focused on facilitating more reliable and reproducible research in the realm of Web measurement and privacy. Yadav et al. (2015) studied a set of open-source web crawlers and showcased how each was suitable for different use cases. More recently, Ahmed et al. (2020) showed the impact that different crawlers had on measurement and security research inferences. Along similar lines, Zeber et al. (2020) and Jueckstock et al. (2021) also showed how the choice of crawler and configuration could harm the repeatability of an experiment. Our work extends these efforts by identifying platform-specific audit challenges.

7 Concluding Remarks

Broader perspectives. This study broadly improves our understanding of how audit configurations can influence our understanding of the behavior of recommendation algorithms. This understanding is expected to become especially important in the coming years because of the growing conversations relating to regulating AI, content recommendation systems, and online platforms (WH2 2022). In the more immediate future, we expect that the findings and recommendations contained within our work will lead to an improvement of the reliability and accuracy of YouTube audit studies, while simultaneously introducing context around (seemingly) contradictory inferences. Other platforms, such as e-commerce and economy sharing platforms, may differ in their interactions. However, sock-puppet approaches are still common ways to study their recommendation systems. Therefore, our high-level approach of configuration-specific investigations can be used to identify cost-saving mechanisms for their audits. It should be noted that the recommendations made in this paper are specific to the YouTube platform. One platform-independent recommendation that our study suggests is that auditors carefully detail the configurations associated with their audits to prevent confusion regarding algorithmic behaviors and to facilitate reproducibility. We do not anticipate harmful societal consequences as a result of our research. However, there are associated limitations and ethical considerations which we outline below.

Limitations. Fundamentally, our work is a best-effort study to understand the impact of different audit methodological decisions on the characteristics of content recommended by YouTube — one of the most commonly audited online platforms. Thus, our study is not without limitations. First, we ourselves are computationally and economically limited and had to make decisions about crawl parameters to explore. This impacted our ability to (1) perform exploration of more paths in each recommendation tree; (2) conduct more than eight synchronized tree explorations; and (3) explore recommendation trees to a greater depth. We mitigate any incorrect inferences that might result from these limitations by only performing like-for-like node- and position-dependent comparisons and ensuring that any differences measured in our study account for the general probabilistic nature of the recommendation algorithm by measuring across-group differences and comparing them with within-group differences. Second, there are latent effects that cannot be controlled from our external vantage point which is effectively measuring a black-box system. We do our best to identify several of these (e.g., A/B testing, data center location, measurement location, etc.) and attempt to counter each of them. However, it is possible that unaccounted effects might still impact our results. Finally, we acknowledge that our choice of a training set and seed video might ultimately not be sufficient to observe all effects of interactions on the recommendation system. Regardless, we provide useful data points for consideration to a community grappling with an ever-growing list of contradictory results.

Ethical considerations. Our study involves crawling and scraping of the YouTube platform, which necessitates ethical considerations along two dimensions: platform costs and privacy harms. *Platform costs.* Our study does interact with the YouTube platform, servers, and content. However, the costs incurred by YouTube from our measurements were minimal and no more than necessary to complete our analysis with statistical rigor. Specifically, our study ensured this with four decisions. First, our study leveraged the official YouTube API whenever possible and used a fully-fledged web browser only when the API did not provide the data required for our study (e.g., recommended videos). Second, the costs incurred by YouTube from our crawls are expected to be insignificant since we only viewed $\approx$ 4.5K videos over the duration of this study. Third, when YouTube accounts were required by our study we purchased accounts directly from Google via a Google Workspace account (at $8/month/account). Finally, whenever possible our crawlers skipped any displayed ads and did not actively interact (beyond viewing) with content or users encountered during the crawl. Thus, our influence on the ad and recommendations algorithms were minimized. Overall, our crawls were in line with typical auditing studies as described in (Sandvig et al. 2014) and are legally permissible (Addicks 2022; Berzon 2022).

Privacy harms. Our study does not involve any human subjects or gather any personally identifiable information. In our public release, all videos are only listed by their URL — therefore, they automatically respect the privacy choices of their associated creators (unlisting the video will make our link inactive). Overall, we respect the principle of beneficence as outlined by the Belmont report (Beauchamp 2008).

Conclusions. This work showcased the effect of audit configurations on the characteristics of recommendation trees generated by them. Specifically, we showed that although training sets do have a statistical impact on recommendations, their effects can be significantly dampened by a 'recency bias' in YouTube's recommendations (§3). Therefore, specific care needs to be taken when selecting videos to view in an audit. More importantly, these decisions need to be disclosed and any audit inferences *must* be conditioned on them. Our analysis of different types of auditing profiles (§4) showed that the expensive task of obtaining clean YouTube accounts would not yield significantly different outcomes than simply maintaining the YouTube cookie for the entire duration of an audit. Further, our findings also suggest that account reuse can be possible by using the 'clear history' feature provided by YouTube. Finally, our analyses of various computational compromises in audits (§5) show that audits do not need to watch a specific fraction of a video for it to impact subsequent recommendations (rather, a preset threshold appears sufficient), challenging automation tasks such as programming cursor clicks on videos do not need to be performed by auditors, and that the depth of a crawl can impact characteristics of the recommendation tree (and should therefore be used to condition any reported inferences from audits).

Acknowledgements

The authors are grateful to the reviewers. This research was supported by the US Air Force Office of Scientific Research under award #9550-20-1-0346 and the US Department of Defense under award #9550-23-4-0730. The opinions here are not necessarily endorsed by the funding bodies.

References

2022. Readout of White House Listening Session on Tech Platform Accountability. Technical report, The White House.

Addicks, M. 2022. Van Buren v. United States: The Supreme Court's Ruling on the Fate of Web Scraping-" Access" to Discovery or Detention? *Tul. J. Tech. & Intell. Prop.*, 24.

Ahmad, S. S.; Dar, M. D.; Zaffar, M. F.; Vallina-Rodriguez, N.; and Nithyanand, R. 2020. Apophanies or Epiphanies? How Crawlers Impact Our Understanding of the Web. WWW '20. New York, NY, USA: Association for Computing Machinery.

Barberá, P. 2020. Social media, echo chambers, and political polarization. *Social media and democracy: The state of the field, prospects for reform*, 34.

Beauchamp, T. L. 2008. The belmont report. *The Oxford textbook of clinical research ethics*, 149–155.

Berzon, J. 2022. Hiq Labs, Inc.. V. LINKEDIN Corporation, no. 17-16783 (9th cir. 2022).

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*

Bryant, L. V. 2020. The YouTube algorithm and the alt-right filter bubble. *Open Information Science*, 4(1): 85–90.

Chen, A. Y.; Nyhan, B.; Reifler, J.; Robertson, R. E.; and Wilson, C. 2023. Subscriptions and external links help drive resentful users to alternative and extremist YouTube channels. *Science Advances*, 9(35): eadd8080.

Covington, P.; Adams, J.; and Sargin, E. 2016. Deep Neural Networks for YouTube Recommendations. RecSys '16. Association for Computing Machinery. ISBN 9781450340359.

DiCiccio, T. J.; and Efron, B. 1996. Bootstrap confidence intervals. *Statistical science*, 11(3): 189–228.

Dunna, A.; Keith, K.; Zuckerman, E.; Vallina-Rodriguez, N.; O'Connor, B.; and Nithyanand, R. 2022. Paying Attention to the Algorithm Behind the Curtain: Bringing Transparency to YouTube's Demonetization Algorithms. In *Proceedings of the 2022 ACM Conference on Computer Supported Cooperative Work (ACM CSCW 2022).*

Efron, B. 1987. Better bootstrap confidence intervals. *Journal of the American statistical Association*, 82(397): 171–185.

Facebook. 2022. About A/B Testing. https://www.facebook.com/business/help/1738164643098669?id=445653312788501. Accessed: 2023-05-10.

FORCE11. 2020. The FAIR Data principles. https://force11.org/info/the-fair-data-principles/. Accessed: 2023-05-12.

Fu, B.; Chi, E.; Cao, P.; Yang, R.; and Singh, S. 2016. Video WatchTime and Comment Sentiment: Experience from YouTube.

Funk, M. 2020. How Does YouTube Count Views? It's more tricky than you think! Tubics.

Gebru, T.; Morgenstern, J.; Vecchione, B.; Vaughan, J. W.; Wallach, H.; Iii, H. D.; and Crawford, K. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92.

Goodman, E. P.; and Trehu, J. 2023. ALGORITHMIC AUDITING: CHASING AI ACCOUNTABILITY. *Santa Clara High Technology Law Journal*, 39(3): 289.

Hannak, A.; Sapiezynski, P.; Molavi Kakhki, A.; Krishnamurthy, B.; Lazer, D.; Mislove, A.; and Wilson, C. 2013. Measuring Personalization of Web Search. WWW '13. Association for Computing Machinery.

Haroon, M.; Chhabra, A.; Liu, X.; Mohapatra, P.; Shafiq, Z.; and Wojcieszak, M. 2022. YouTube, The Great Radicalizer? Auditing and Mitigating Ideological Biases in YouTube Recommendations. *arXiv preprint arXiv:2203.10666*.

Hosseinmardi, H.; Ghasemian, A.; Clauset, A.; Mobius, M.; Rothschild, D. M.; and Watts, D. J. 2021. Examining the consumption of radical content on YouTube. *Proceedings of the National Academy of Sciences*, 118(32): e2101967118.

Hussein, E.; Juneja, P.; and Mitra, T. 2020. Measuring misinformation in video search platforms: An audit study on YouTube. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1): 1–27.

Ibrahim, H.; AlDahoul, N.; Lee, S.; Rahwan, T.; and Zaki, Y. 2023. YouTube's recommendation algorithm is left-leaning in the United States. *PNAS nexus*, 2(8): pgad264.

Jueckstock, J.; Sarker, S.; Snyder, P.; Beggs, A.; Papadopoulos, P.; Varvello, M.; Livshits, B.; and Kapravelos, A. 2021. Towards Realistic and ReproducibleWeb Crawl Measurements. In *Proceedings of the Web Conference 2021*, WWW '21. Association for Computing Machinery.

Khattak, S.; Fifield, D.; Afroz, S.; Javed, M.; Sundaresan, S.; McCoy, D.; Paxson, V.; and Murdoch, S. J. 2016. Do You See What I See? Differential Treatment of Anonymous Users. In *NDSS*.

Ledwich, M.; and Zaitsev, A. 2019. Algorithmic Extremism: Examining YouTube's Rabbit Hole of Radicalization. *CoRR*, abs/1912.11211.

Lutz, M.; Gadaginmath, S.; Vairavan, N.; and Mui, P. 2021. Examining Political Bias within YouTube Search and Recommendation Algorithms. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1–7. IEEE.

Makhortykh, M.; and Urman, A. 2020. The Great Randomizer: Using Virtual Agents For Auditing The Effects Of Youtube Recommendation Algorithm On Ideologically-Charged News Content Distribution. *AoIR Selected Papers of Internet Research*.

Metaxa, D.; Park, J. S.; Robertson, R. E.; Karahalios, K.; Wilson, C.; Hancock, J.; Sandvig, C.; et al. 2021. Auditing algorithms: Understanding algorithmic systems from the outside in. *Foundations and Trends® in Human–Computer Interaction*, 14(4): 272–344.

Munger, K.; and Phillips, J. 2022. Right-wing YouTube: A supply and demand perspective. *The International Journal of Press/Politics*, 27(1): 186–219.

Papadamou, K.; Papasavva, A.; Zannettou, S.; Blackburn, J.; Kourtellis, N.; Leontiadis, I.; Stringhini, G.; and Sirivianos, M. 2019. Disturbed YouTube for Kids: Characterizing and Detecting Disturbing Content on YouTube. *CoRR*, abs/1901.07046.

Papadamou, K.; Zannettou, S.; Blackburn, J.; Cristofaro, E. D.; Stringhini, G.; and Sirivianos, M. 2020. Understanding the Incel Community on YouTube. *CoRR*, abs/2001.08293.

Papadamou, K.; Zannettou, S.; Blackburn, J.; Cristofaro, E. D.; Stringhini, G.; and Sirivianos, M. 2021. "It is just a flu": Assessing the Effect of Watch History on YouTube's Pseudoscientific Video Recommendations.

Parsons, J. 2017. How Long Until Watching a YouTube Video Counts as a View? GrowTraffic.

Ram, M.; and Davim, J. P. 2018. *Advanced mathematical techniques in engineering sciences*, 149–164. CRC Press.

Reimers, N.; and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ribeiro, M. H.; Ottoni, R.; West, R.; Almeida, V. A. F.; and Meira, W. 2020. Auditing Radicalization Pathways on YouTube. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. New York, NY, USA: Association for Computing Machinery.

Ribeiro, M. H.; Veselovsky, V.; and West, R. 2023. The Amplification Paradox in Recommender Systems. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 17, 1138–1142.

Rosario, B. 2000. Latent semantic indexing: An overview. *Techn. rep. INFOSYS*.

Sandvig, C.; Hamilton, K.; Karahalios, K.; and Langbort, C. 2014. Auditing algorithms: Research methods for detecting discrimination on internet platforms. *Data and discrimination: converting critical concerns into productive inquiry*, 22(2014): 4349–4357.

Song, K.; Tan, X.; Qin, T.; Lu, J.; and Liu, T.-Y. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33: 16857–16867.

SpaCy. 2022. SpaCy linguistic features. https://spacy.io/usage/linguistic-features. Accessed: 2023-09-10.

Tang, J.; Belletti, F.; Jain, S.; Chen, M.; Beutel, A.; Xu, C.; and H. Chi, E. 2019. Towards Neural Mixture Recommender for Long Range Dependent User Sequences. WWW '19. New York, NY, USA: Association for Computing Machinery.

Tomlein, M.; Pecher, B.; Simko, J.; Srba, I.; Móro, R.; Stefancova, E.; Kompan, M.; Hrckova, A.; Podrouzek, J.; and Bieliková, M. 2021. An Audit of Misinformation Filter Bubbles on YouTube: Bubble Bursting and Recent Behavior Changes. In *RecSys '21: Fifteenth ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September 2021 - 1 October 2021*.

Yadav, M.; and Goyal, N. 2015. Comparison of Open Source Crawlers-A Review.

Zeber, D.; Bird, S.; Oliveira, C.; Rudametkin, W.; Segall, I.; Wollsén, F.; and Lopatka, M. 2020. The Representativeness of Automated Web Crawls as a Surrogate for Human Browsing. In *Proceedings of The Web Conference 2020*, WWW '20. New York, NY, USA: Association for Computing Machinery.

Zhao, Z.; Cheng, Z.; Hong, L.; and Chi, E. H. 2015. Improving User Topic Interest Profiles by Behavior Factorization. WWW '15.

Zhao, Z.; Hong, L.; Wei, L.; Chen, J.; Nath, A.; Andrews, S.; Kumthekar, A.; Sathiamoorthy, M.; Yi, X.; and Chi, E. 2019. Recommending What Video to Watch next: A Multitask Ranking System. RecSys '19. New York, NY, USA: Association for Computing Machinery.

8 Ethics Checklist

1. For most authors...
   - (a) Would answering this research question advance science without violating social contracts, such as violating privacy norms, perpetuating unfair profiling, exacerbating the socio-economic divide, or implying disrespect to societies or cultures? Yes, see Ethical Considerations section.
   - (b) Do your main claims in the abstract and introduction accurately reflect the paper's contributions and scope? Yes
   - (c) Do you clarify how the proposed methodological approach is appropriate for the claims made? Yes, we explain the validity of our approach in the Methodology section
   - (d) Do you clarify what are possible artifacts in the data used, given population-specific distributions? Yes, We identify the these artifacts as latent effects in data gathering section
   - (e) Did you describe the limitations of your work? Yes, we have a dedication limitations section
   - (f) Did you discuss any potential negative societal impacts of your work? Yes, see Ethical Considerations section
   - (g) Did you discuss any potential misuse of your work? No, our work recommends certain practices while auditing platforms, these recommendations can be ignored but not misused.
   - (h) Did you describe steps taken to prevent or mitigate potential negative outcomes of the research, such as data and model documentation, data anonymization, responsible release, access control, and the reproducibility of findings? Yes, our data does not include any privacy sensitive data points, so we have shared the videos used in the training phase. See Methodology for the url.
   - (i) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes
2. Additionally, if your study involves hypotheses testing...
   - (a) Did you clearly state the assumptions underlying all theoretical results? NA
   - (b) Have you provided justifications for all theoretical results? NA
   - (c) Did you discuss competing hypotheses or theories that might challenge or complement your theoretical results? NA
   - (d) Have you considered alternative mechanisms or explanations that might account for the same outcomes observed in your study? NA
   - (e) Did you address potential biases or limitations in your theoretical framework? NA
   - (f) Have you related your theoretical results to the existing literature in social science? NA
   - (g) Did you discuss the implications of your theoretical results for policy, practice, or further research in the social science domain? NA
3. Additionally, if you are including theoretical proofs...
   - (a) Did you state the full set of assumptions of all theoretical results? NA
   - (b) Did you include complete proofs of all theoretical results? NA
4. Additionally, if you ran machine learning experiments...
   - (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? NA
   - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? NA
   - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? NA
   - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? NA
   - (e) Do you justify how the proposed evaluation is sufficient and appropriate to the claims made? NA
   - (f) Do you discuss what is "the cost" of misclassification and fault (in)tolerance? NA
5. Additionally, if you are using existing assets (e.g., code, data, models) or curating/releasing new assets, **without compromising anonymity**...
   - (a) If your work uses existing assets, did you cite the creators? NA
   - (b) Did you mention the license of the assets? NA
   - (c) Did you include any new assets in the supplemental material or as a URL? NA
   - (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? NA
   - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? NA
   - (f) If you are curating or releasing new datasets, did you discuss how you intend to make your datasets FAIR (see (FORCE11 2020))? NA
   - (g) If you are curating or releasing new datasets, did you create a Datasheet for the Dataset (see (Gebru et al. 2021))? NA
6. Additionally, if you used crowdsourcing or conducted research with human subjects, **without compromising anonymity**...
   - (a) Did you include the full text of instructions given to participants and screenshots? NA
   - (b) Did you describe any potential participant risks, with mentions of Institutional Review Board (IRB) approvals? NA
   - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? NA
   - (d) Did you discuss how data is stored, shared, and deidentified? NA