ConvEx: A Visual Conversation Exploration System for Discord Moderators

FREDERICK CHOI, TANVI BAJPAI, SOWMYA PRATIPATI, and
ESHWAR CHANDRASEKHARAN, University of Illinois at Urbana-Champaign, USA

Moderators are at the core of maintaining healthy online communities. For these moderators, who are often volunteers from the community, filtering through content and responding to misbehavior on time has become increasingly challenging as online communities continue to grow. To address such challenges of scale, recent research has looked into designing better tools for moderators of various platforms (e.g. Reddit, Twitch, Facebook, and Twitter). In this paper, we focus on Discord, a platform where communities are typically involved in large, synchronous group chats, creating an environment with a faster pace and a lack of structure compared to previously studied platforms. To tackle the unique challenges presented by Discord, we developed a new human-AI system called *ConvEx* for exploring online conversations. ConvEx is an AI-augmented version of the standard Discord interface designed to help moderators be proactive in identifying and preventing potential problems. It provides visual embeddings of conversational metrics, such as activity and toxicity levels, and can be extended to visualize other metrics. Through a user study with eight active moderators of Discord servers, we found that ConvEx supported several high-level strategies in monitoring a server and analyzing conversations. ConvEx allowed moderators to obtain a holistic view of activity across multiple channels on the server while guiding their attention towards problematic conversations and messages in a channel, helping them identify important contextual information to obtain reliable information from the AI analysis while also being able to pick up on contextual nuances which the AI missed. We conclude with design considerations for integrating AI into future interfaces for moderating synchronous, unstructured online conversations.

CCS Concepts: • **Human-centered computing → Collaborative and social computing systems and tools**.

Additional Key Words and Phrases: online moderation; mixed-initiative; socio-technical systems; open source

ACM Reference Format:
Frederick Choi, Tanvi Bajpai, Sowmya Pratipati, and, Eshwar Chandrasekharan. 2023. ConvEx: A Visual Conversation Exploration System for Discord Moderators. *Proc. ACM Hum.-Comput. Interact.* 7, CSCW2, Article 262 (October 2023), 30 pages. https://doi.org/10.1145/3610053

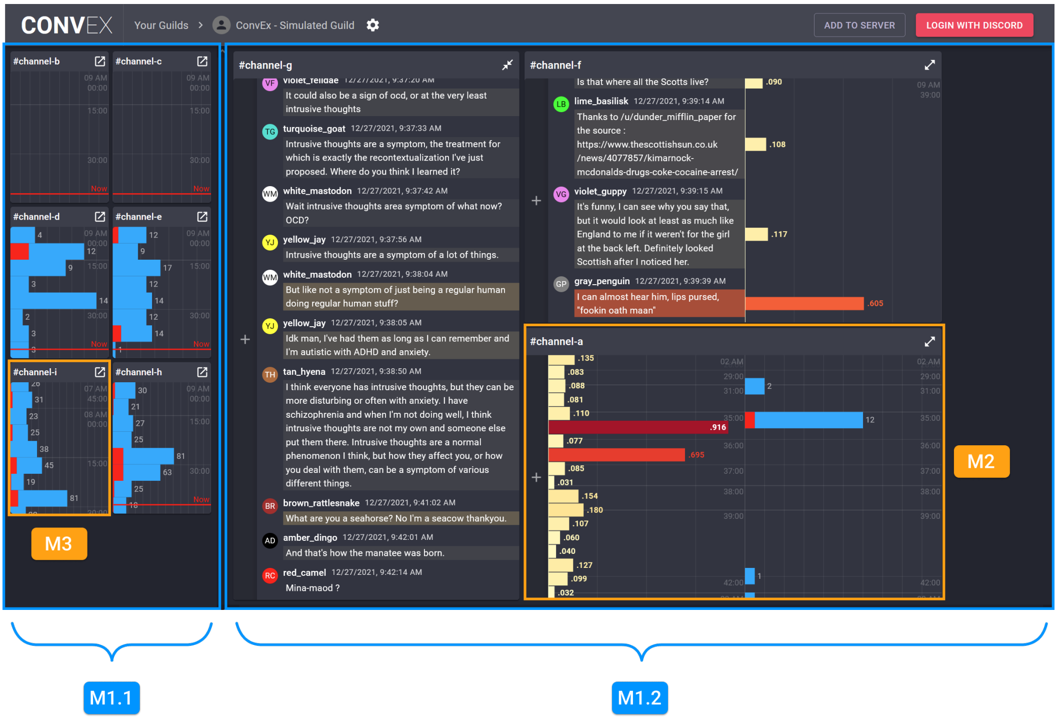
1 INTRODUCTION

> *"Moderators are a key part of making communities great and a place where people want to gather." - Discord Moderator Academy [1]*

Online communities and their worldwide growth have provided us with countless benefits. They connect people across geographical boundaries and cultural differences. They provide spaces for those who identify as LGBTQ+ to interact, learn from, and help each other where it may be unsafe to do so elsewhere [48]. At the heart of these online spaces are the moderators who play a critical

Authors' address: Frederick Choi; Tanvi Bajpai; Sowmya Pratipati;
Eshwar Chandrasekharan, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA.

2573-0142/2023/10-ART262 $15.00
https://doi.org/10.1145/3610053



violet_felidae 12/27/2021, 9:37:20 AM
It could also be a sign of ocd, or at the very least intrusive thoughts

turquoise_goat 12/27/2021, 9:37:33 AM
Intrusive thoughts are a symptom, the treatment for which is exactly the recontextualization I've just proposed. Where do you think I learned it?

white_mastodon 12/27/2021, 9:37:42 AM
Wait intrusive thoughts area symptom of what now? OCD?

yellow_jay 12/27/2021, 9:37:56 AM
Intrusive thoughts are a symptom of a lot of things.

white_mastodon 12/27/2021, 9:38:04 AM
But like not a symptom of just being a regular human doing regular human stuff?

yellow_jay 12/27/2021, 9:38:05 AM
Idk man, I've had them as long as I can remember and I'm autistic with ADHD and anxiety.

tan_hyena 12/27/2021, 9:38:50 AM
I think everyone has intrusive thoughts, but they can be more disturbing or often with anxiety. I have schizophrenia and when I'm not doing well, I think intrusive thoughts are not my own and someone else put them there. Intrusive thoughts are a normal phenomenon I think, but how they affect you, or how you deal with them, can be a symptom of various different things.

brown_rattlesnake 12/27/2021, 9:41:02 AM
What are you a seahorse? No I'm a seacow thankyou.

amber_dingo 12/27/2021, 9:42:01 AM
And that's how the manatee was born.

red_camel 12/27/2021, 9:42:14 AM
Mina-maod ?

Is that where all the Scotts live?

lime_basilisk 12/27/2021, 9:39:14 AM
Thanks to /u/dunder_mifflin_paper for the source : https://www.thescottishsun.co.uk /news/4077857/kimarnock-mcdonalds-drugs-coke-cocaine-arrest/

violet_guppy 12/27/2021, 9:39:15 AM
It's funny, I can see why you say that, but it would look at least as much like England to me if it weren't for the girl at the back left. Definitely looked Scottish after I noticed her.

gray_penguin 12/27/2021, 9:39:39 AM
I can almost hear him, lips pursed, "fookin oath maan"

Fig. 1. ConvEx provides an AI-augmented interface designed to help moderators monitor a Discord server with different levels of oversight. Shown here is the main interface and all of its components. The interface is split into two major regions, labeled here in this image: M1.1) channel palette, M1.2) workspace. The channel palette contains tiles that represent individual channels within a server and include a high-level summary of activity within that channel (e.g., M3). Dragging a tile over to the workspace area reveals a more detailed view of messages in the channel and presents more options for visualization (e.g., M2). ConvEx is designed to help moderators monitor activity across multiple channels, and guide their attention towards specific conversations/messages when needed.

role in maintaining an online community and keeping its members safe [20]. Moderators have to contend with behavior that can easily disrupt the social benefits of online communities [49], including toxicity and hate speech, which is disproportionately targeted at LGBTQ+ identities [48], women [44, 56], youth [17], and other underrepresented users [57]. Other challenges include raids, where a sudden, typically large, influx of users (or bots) enter a community and interact with no intention of staying. These are disruptive and make a moderator's job more difficult [32]. As online communities grow rapidly, the amount of traffic outpaces the capacity of the moderators who, despite often being unpaid volunteers [40], shoulder the responsibilities of reviewing flagged content and monitoring interactions [58].

1.1 Moderation Challenges on Discord

In this paper, we focus on Discord and the challenges faced by volunteer moderators on this platform. Discord is unique as a social media platform, home to servers (also referred to as guilds) with anywhere from just one member to more than a million members who can hop into one of several text or voice channels and type or talk with each other. As a group chat platform, Discord organizes text-based interactions in channels, where messages are presented in a list-like manner and older messages are pushed up and out of view as newer messages get posted. Other than an optional reply feature that indicates a new message is a response to a previously posted message, Discord channels lack the structure that helps to delineate content. Platforms such as Twitter or Reddit impose this structure by way of post or comment-based framing which clearly indicates relationships and distinctions between units of content. Despite the existence of a reply feature on Discord, users can ultimately just reply by sending another message to the chat, making it difficult to identify related messages and understand the context of any one message. The interface of Discord itself becomes an obstacle as it creates blind spots for moderators. The synchronous nature of the interactions on Discord in both its text and voice channels demands constant attention across the whole server, but only one channel can be viewed at a time. While a moderator is reading through one channel, they are effectively blind to any activity in all the other channels [32]. As servers typically have dozens of channels with varying levels of activity within each, the current interface only affords a narrow view of all the activity within a server. Moderators also have to moderate voice channels, which present a different set of challenges, further reducing the amount of time and attention they can spend on text-only channels. As such, moderators have adopted unique strategies that require novel tools and automation techniques.

1.2 Challenges in Using AI for Moderation

Recent research in machine-learning [25, 50, 62] has shown promising advances toward using computation to reduce moderator workload and make content moderation more effective at larger scales. Natural language processing in particular has seen an explosion of progress in detecting hate speech and toxicity [2, 16] and in predicting the short-term trajectory of behavior [5, 15, 22, 61], all with increasing accuracy in the last decade. Unfortunately, these are still largely not yet suitable for automating most moderation tasks as these algorithms lack a nuanced, contextual understanding of harm [33, 50]. However, we believe that such algorithms can still provide value to moderators when incorporated into an appropriate, human-in-the-loop system. In this work, we investigate the following through our design and evaluation of ConvEx:

> *If state-of-the-art models are not yet suitable for fully automating moderation tasks, what are effective methods of incorporating their outputs into an interface for moderators to reduce their workload?*

1.3 Introducing ConvEx

In this paper, we present a new system, called ConvEx (short for Conversation Explorer), to explore conversations by augmenting the standard Discord interface with visual embeddings of conversational metrics like activity and toxicity levels. By computationally condensing information and adding structure to chat, our goal is to reduce the cognitive load for moderators and expedite their overall workflow. The user-interface and visualizations were iteratively designed through a pilot study.

The final system as described in this paper was evaluated in a user-study with eight active moderators of Discord servers. We observed that moderators were able to effectively use both message-level and discourse-level metrics visualized by ConvEx. Single-point metrics of individual messages were rendered as a heatmap that moderators could easily interpret and use to identify specific messages to focus their attention on. On the other hand, the visualization of discourse-level metrics from aggregating the single-point analyses of several messages in a row highlighted the overall trajectory of a conversation. This allowed moderators to passively monitor several channels within the server simultaneously, narrow their attention to the most important channels, and locate problematic conversations and messages within a channel quickly.

1.4 Summary of Contributions

With ConvEx, we aim to bridge the gap between recent advances in AI techniques and moderation tools by implementing a human-AI system that makes state-of-the-art algorithms accessible to moderators working with unstructured chat. We also present a novel mixed-initiative interaction design based on the presentation of AI predictions that allows for flexible interpretation. Our design not only reduces the cost of analysis but also reduces the cost of error as the user implicitly incorporates their own contextual knowledge when interpreting the output. We plan to release ConvEx as an open-source application for anyone to deploy on their servers.[1] In the future, ConvEx can be extended to visualize other conversational metrics, including those derived from non-textual features.

The contributions in this paper are three-fold. First, we present five key design objectives to guide the design of AI assisted moderation tools. Next, we implement these design guidelines into practice to build a new human-AI system for exploring online conversations called *ConvEx*. Finally from our design, implementation, and user studies, we develop additional guidelines and considerations for designers of future systems that aim to facilitate moderation at scale with computational assistance, and how to make responsible and effective use of AI algorithms given their imperfect state in the present day.

2 BACKGROUND

Now, we review related work on existing moderation approaches and the role of AI in moderation. We defer the discussion of prior work regarding moderation on Discord and moderation tools for other social platforms to Section 3, since they will be used to synthesize the five primary design goals utilized in developing ConvEx.

2.1 Current State of Moderation

Research has consistently shown that women, youth, racial minorities, and members of the LGBTQ+ community are disproportionately the target of harm and harassment online [42, 44, 48, 49, 56]. And platforms, moderators, and communities have developed varied strategies for mitigating these harms. Some approaches are directed at sanctioning and removing bad actors. In the mid-2010s, the rise of highly problematic communities on Reddit known for their tendency to spread hate speech and misinformation led to several subreddits being banned or quarantined. This presented an opportunity to research the effects of such interventions on the spread of hate speech and misinformation on Reddit [11, 12]. Also of interest was how users and/or communities may have migrated in response to the subreddit bans [43], conflicts that emerge between communities [36], and whether deplatforming has been successful in mitigating harm when regarding the internet as a whole [3, 47]. Bad actors have been shown to employ their own strategies to counter moderation efforts, such as how some users started to use lexical variants and intentional misspellings to circumvent the banning of tags on Instagram that promoted the practice of harmful eating disorders [9]. In the case of shared blocklists, researchers have observed how the same strategy used to mitigate harm has the potential to be hijacked and become the cause of harm instead [31].

Researchers have long peered into the strategies that moderators employ [7, 19, 41, 51, 53], examining the effects of feedback to individual users [14, 29], as well as platform interventions such as community-level quarantines [11, 21] and bans [12, 23, 37]. Research has shown that the effects of feedback from the community are varied, with negative feedback possibly encouraging bad behavior [14], while some forms of discouragement are more or less effective in mitigating bad behavior [52]. While some suggest transparency in moderation leads to better adherence to

[1] https://github.com/koreanwglasses/ConvEx-Demo

rules [35], the empirical effect of transparency in moderation is equivocal [29]. The role of norms and example setting is observed across communities of diverse scales [13, 52]. Research has also investigated the factors that affect the stability of norms once set, for better or for worse [45].

Managing increasing scale and volume is reported again and again as one of the biggest challenges for moderators of growing online communities [28, 34, 39, 46]. And as the landscape of online social spaces continues to evolve, so do techniques that moderators use. As an example, many pre-existing communities on Reddit decided to expand to Discord, a relatively new platform, and many of their moderators oversaw their community on both platforms [34]. Moderators might be able to carry over some tactics, perhaps, by writing user-scripts and bots that emulate tools they are used to [34], but every new platform inevitably brings with it completely new challenges [32]. And so, moderators' workflows have evolved with the changes to their platform and the tools they use to keep up [53]. Twitch is a notable example, as the synchronous nature of its streams and chat parallels Discord's voice and text channels, and researchers have investigated how moderators' strategies adapted to the tools that Twitch provides [6, 7]. In addition to tools that are provided by the platform itself, moderators often develop their own tools [6, 28, 34, 53] which are sometimes shared for others to use, even reaching ubiquity in the case of Reddit's AutoModerator [28]. But considering that platforms like Reddit, and Discord rely primarily on unpaid, volunteer moderators to maintain their communities, the rapid growth of platforms has widened the gap between the amount of work that moderators put in and the support they receive [40].

Research into the space of automated tools used by moderators of online communities reveals that keyword and regex matching are the most common approaches to partially automating the review of text-based content [28, 34]. But these tools seem outdated when compared to the amount of research that focuses on computational models that predict various measures of conversation health, including toxicity [2], presence of personal attacks [59], controversiality [22], and conversation derailment [5]. There is a critical need for research into the design and integration of such models into the moderation toolkit on actual platforms. This has been difficult, partly due to the care that must be taken to avoid imparting harmful biases onto the moderator and their community [8, 33]. We seek a successful integration of such technology into the moderator workflow that requires a sensitive approach that factors in the needs of the platform, the community, and the moderators, all while accounting for the inaccuracies inherent in emerging technologies.

2.2 AI in Moderation

The last decade has seen leaps in the automated detection of problematic online behavior. Techniques range from surface-level features including character or word embeddings and linguistic features to, more recently, features that capture semantic and contextual meaning, including sentiment, and knowledge-based models [50]. Non-textual features such as images have also been used to train models to detect cyberbullying [25, 62]. Researchers have developed models that detect problematic content, including toxicity, hate speech, and offensive language [2, 16], as well as behavioral warning signs, such as early signs of conversational failure [61] and the predictive detection of controversy-causing posts [22]. Researchers have also developed models that predict positive behavior, such as politeness [15] and prosociality [5].

There have been a handful of mixed-initiative approaches to incorporating state-of-the-art detection into end-user tools for content moderation. Notably, Crossmod [10], designed for Reddit moderators, presents an interface where moderators can configure automatic actions based on AI models that predict the violation of various macro-norms within posts. The actions include automatic removal as well as triaging, which allow a human moderator to manually review a post. Another approach is Synthesized Social Signals [27], designed for use by individuals on Twitter, which displays the output of a model that scores the toxicity of another user, and another model that scores their tendency to spread misinformation. These are displayed as icons that are displayed next to a user's profile picture and are designed to help an individual judge the trustworthiness of a source as they browse, and assess the potential cost of interaction.

The next breakthrough in moderation lies in the use of algorithmic techniques to help moderators manage scale. However, the primary challenge is ensuring that any system that employs such techniques is fair. While NLP models that analyze text for problematic behavior are promising, the problem is far from solved, and the varying definitions of hate speech by different authors and the narrow contexts for which different techniques were developed [50] make it difficult to rely on any one model for all use cases. Even with a human-in-the-loop, it is not trivial to design a system that is accurate and reliable. For example, feature-based explanations of toxicity predictions yield no significant improvement in helping humans detect misclassifications [8]. As algorithmic predictions have a significant influence on a moderator's decisions and workflow in nuanced ways [55], it is critical that care is taken into designing even mixed-initiative systems that employ automated detection. An ethical framework for automated moderation [33], understanding stakeholder values [54], and involving the community in the process [60] are among the latest strategies researchers have proposed for building fair and ethical systems.

3 DESIGN OBJECTIVES

In this section, we introduce the five primary design objectives used in developing our conversation explorer, ConvEx. These design objectives are informed by previous research related to moderation and moderation tools on Discord and other platforms. As such, we will discuss this related work in tandem with each design goal it motivates.

Previous research on moderation in Discord servers revealed that one of the major challenges is being able to catch and act on bad behavior quickly [34]. Unlike posting on a forum-like platform such as Reddit, instant messaging on a platform like Discord is synchronous. As a result, moderators have limited time to respond to bad behavior before the conversation moves on and the full extent of harm has been done. However, on large servers with several active channels, it can be difficult to keep track of all the activity at once, especially since Discord only shows the user one channel at a time. Other than user-submitted reports from the community, there is no way to keep an eye on a channel without having it open. Effectively, this means that when a moderator is focusing on one channel, the rest of the channels become giant blind spots where problematic activity can go unnoticed. And so, instead of having to click through various channels, the moderator should be able to get a holistic view of the activity across their entire server, so they can effectively monitor multiple conversations happening at the same time.

> ***Design Goal 1 (G1).*** *ConvEx should help moderators obtain a holistic view of activity across multiple channels and help them monitor multiple conversations simultaneously.*

We can start by making all the messages in all the channels visible at the same time. Having a full view of the whole server instead of just a single channel at a time gives a much broader view of the whole server at once. However, for larger servers with a high level of activity, the volume of messages being sent across all the different channels can easily exceed the average human reading speed, reducing the effectiveness of such an approach.

Prior work on moderation has studied different approaches to manage scale—e.g., growth in community size and overall activity levels—across several platforms [10, 26, 28, 30]. Lampe et al. [39] studied a form of community-supported distributed moderation on Slashdot, where the work of moderation is distributed across community members who review and vote on comments. However, even with the distribution of work, many comments remained unseen and unmoderated [18]. Purely algorithmic approaches are tempting as algorithms scale much more easily than humans do, but even state-of-the-art natural language processing algorithms struggle to detect the nuances of humor, sarcasm, and context. Human-machine collaboration approaches have also been studied [10, 27, 28], where algorithms do part of the work of parsing through a large body of content, surfacing content that needs immediate attention, and leaving the final decision in the hands of the human moderators. On Reddit, Automod [28] is a popular tool in the human-machine collaboration category, where moderators can set up regular expressions to automatically detect some violations of their rules.

On Discord, Dyno and Carl.gg are the most used bots on discord and are the first choice for creators of new servers, as they feature basic spam detection and keyword-based filtering. However, in general, there is a lack of moderation tools for Discord servers. Moderators sometimes write their own bots which are often specialized for their own server [34]. Due to various factors, including the desire to keep one's moderation systems opaque from potential bad actors, these often do not find their way into other communities. This can pose a barrier for smaller, newer servers that want to grow but do not have the tools to efficiently manage a larger community.

With more advanced models for toxicity and other language features, these can supersede the basic spam filtering and keyword-based detection algorithms employed in the most widely used moderation tools on Discord. Using more up-to-date sophisticated algorithms with a focus on human-machine collaboration can help moderators by highlighting high-risk areas, allowing them to skim over relatively low-risk areas and reserve their attention for the areas that need it most.

> ***Design Goal 2 (G2).*** *ConvEx should reduce the cognitive burden on moderators by utilizing AI to present conversational metrics that guide moderator attention towards important messages and conversations.*

Since Discord is an instant messaging social platform, text-based conversations are a dominant modality for interaction. Context plays a significantly larger role in this case than it would on a platform like Reddit or Twitter where individual posts are largely self-contained with respect to context or topic. An effective tool for moderation should help facilitate the moderation of content at the discourse level where messages are sent within the implicit context of an exchange (or conversation). However, the lack of structure can make it difficult to quickly identify and understand the context surrounding a particular message or exchange, especially if a lot of messages were exchanged in a short period of time. This is where AI can help augment a moderator's abilities to analyze the context surrounding a conversation. Our tool should help moderators locate and understand the full context surrounding an incident and the users involved.

> ***Design Goal 3 (G3).*** *ConvEx should assist moderators in analyzing and assessing conversations by guiding their attention towards relevant messages and highlighting patterns of activity.*

By emphasizing guiding moderator attention instead of making decisions for them, we can avoid the pitfalls of a fully automated system by ensuring that all final decisions are made following human judgment. This is critical as different communities have widely varying norms [13], and our tool needs the flexibility to be able to adapt to these communities. We can further improve the flexibility of our tool by making it highly configurable. Moderators can thus avoid the overhead of writing specialized, custom-built bots while still having a tool that is suited for their workflow and their community.

> ***Design Goal 4 (G4).*** *ConvEx should be flexible and customizable to adapt to the specific needs of a broad range of communities and moderators.*

The process of learning a new tool in itself is a barrier to adoption and is an overhead we must consider. An example is Automod, a moderation tool for Reddit communities, which requires moderators to take on a new set of tasks related to setting up, configuring, and maintaining the tool [28]. So, although such a tool can help moderators deal with existing challenges, its effectiveness upon deployment relies on moderators taking on a set of new tasks in addition to configuring and maintaining the tool. Moderators already struggle to keep up with the work of moderating a large server and are short-staffed [34]. To minimize the burden of learning a new system, we focus on complementing the existing toolkit instead of trying to replace it. Moderators can then adopt our tool incrementally, and use it to complement their existing tools to passively detect and enforce existing norms.

> ***Design Goal 5 (G5).*** *ConvEx should be easy to learn and compatible with existing servers, focusing on integrating with existing tools instead of trying to replace them.*

We present these design goals as a novel synthesis of guidelines for the design of AI-augmented moderation tools that augment the moderator's abilities and give the moderator full agency over decisions made at every stage. With these design goals to guide our design and implementation, we developed ConvEx, our novel moderation tool which we describe in the following sections.

4 SYSTEM DESIGN

In this paper, we build a new visual conversation exploration tool for Discord called *ConvEx* (Conversation Explorer). ConvEx is a tool for moderators that incorporates AI in a way that is suitable for an instant-messaging platform like Discord. We focus on a visualization-based approach to present moderators with a holistic view of all the text channels in their server (**G1**), using AI models to help moderators filter (**G2**) and comprehend (**G3**) the activity within. We build configurability into our tool (**G4**) and design it to work alongside existing tools (**G5**) so that communities can easily integrate ConvEx into their existing workflows.

4.1 Ethical Considerations

In order to ensure that our tool was built ethically, we made an effort to comply with Discord's terms of service. We use a bot to access messages with our app, as automated requests to the Discord API require a bot token according to Discord's terms of service. We also do not store any message content (or other content from the Discord API)—only the unique identifiers for a message and the results of our NLP analysis are stored. In addition, we use the Discord OAuth API to authenticate our users, allowing us to verify their identity without requiring them to send their credentials directly to our server. We conducted user testing on a simulated Discord server, which was constructed by using Reddit API to fetch messages from the most active posts and obfuscate any usernames within the messages to protect the privacy of Reddit users.

4.2 ConvEx Data Collection and Analysis

To access messages from Discord servers, we built a bot that can access the Discord API[2]. Designed to be minimally invasive, the bot only requires two privileges: reading message data and reading message history. When a moderator adds the ConvEx bot to their server, they are notified of exactly what the bot will need access to (Figure 10, Step 2). Only a single instance of the bot is active at any point, meaning messages from all the servers the bot has been added to are collected in one place. In order to ensure that any messages within a guild are not leaked to outsiders via the ConvEx bot, the moderator must first login before they can see any of their servers. Using the OAuth servers provided by Discord, we are able to securely and reliably determine who the current user is, and determine what servers, channels, and messages they have access to.

[2] https://discord.com/developers/docs/intro

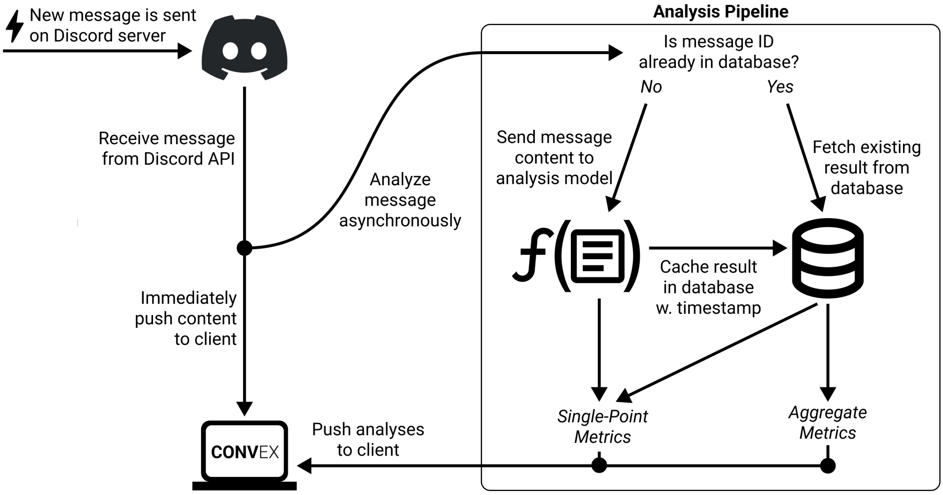

New message is sent on Discord server
Receive message from Discord API
Immediately push content to client
Analyze message asynchronously
CONVEX
Push analyses to client
Analysis Pipeline
Is message ID already in database?
No
Yes
Send message content to analysis model
Fetch existing result from database
Cache result in database w. timestamp
Single-Point Metrics
Aggregate Metrics

Fig. 2. Once ConvEx is authorized on a server, it will listen for any new messages that are sent over that server. The content of the messages is sent to the client to be displayed immediately while the Perspective API computes single-point analyses of each message. The results are stored in a database for quick retrieval at a later point when the client requests previously analyzed messages. Aggregate metrics are computed by querying the database and collecting previously computed toxicity and activity metrics. The analyses are then pushed to the client to be rendered alongside the original message content. The message content is never stored, in compliance with Discord's terms of service, and is instead fetched directly from the Discord API when needed.

Once the moderator is logged in, they are presented with a list of servers/guilds they have added ConvEx to (Figure 10, Step 3). This list is filtered so the moderator only has access to servers that they are a member of. Clicking on any of the server/guild icons will take them to the corresponding guild dashboard (see M1 in Table 1). In order to produce the interactive user features described in the following section, the bot listens for any new messages and pushes them to the appropriate clients via web sockets.

These messages are also sent to the ConvEx backend for automated analysis. We designed ConvEx to be able to use any model or algorithm that takes text as input and returns a prediction probability or summary statistic. As a proof-of-concept, we opted to use the Perspective API[3] as the analysis backend for ConvEx, since it is easily accessible and is a well-known API that has been used for moderation in the past. The Perspective API is used to predict a probability of toxicity given the chat messages as input, though we emphasize that ConvEx can be extended to incorporate models that predict other outcomes such as controversiality or prosocial behavior. We analyze the messages on the server and store the results in a database for faster retrieval when visualizing the results and for computing aggregate statistics from predictions over several messages.

To support navigation through the history of messages sent on a server, the ConvEx backend fetches messages on demand in batches of 100 at a time from the Discord API, which is the maximum for a single API call. The backend then analyzes the content of the messages or fetches previous results from the database (if available), which are then sent to the client's browser for visualization along with the message content itself. We also query the database of analysis results to aggregate statistics over several messages, such as the total number of messages sent and the

[3] https://www.perspectiveapi.com

number of messages with a toxicity score above a fixed threshold. These are computed on messages grouped over intervals of 5-minutes based on the time they were sent, and are are updated as new messages come in or older messages are loaded. Computing these aggregates is fast as it only requires querying previously computed results, and are thus useful for quickly computing high-level statistics on overall activity over time.

4.3 Pilot Study

We built a rudimentary interface on top of the data collection and analysis pipeline, and then conducted a pilot study with three participants on servers they regularly moderate. Following the study, responses from the participants informed our user-interface design as presented in the following section.

Summary of Changes. Initially, all channels defaulted to a full-size heat map view. The feature to customize the layout of the views (I2) was added in response to feedback from pilot study users who indicated that they have a couple of channels they tend to focus on and would want to keep on top. In addition, the channel palette (M1.1) with small charts displaying coarse statistics was added to keep activity from all channels in view at all times. Animations were also reduced from the initial implementation. The pilot study interface would automatically highlight the message the mouse was hovering over and also highlight messages from the same author. This feature was removed following feedback from P1 who noted that the rapid transitions between different filters as one moves the cursor can be distracting. The final implementation of the data pipeline was built and the overall layout of the user interface was redesigned in order to significantly improve the performance of ConvEx for the end user, following observations of all participants struggling to navigate the interface through the long loading times. We present the iterated, post-pilot-study design of the ConvEx interface in the following section.

5 USER INTERFACE DESIGN

The user interface of ConvEx was carefully constructed to present conversational metrics like activity and toxicity levels in an effective way while making efficient use of the data pipeline described in the previous section. Broadly, the features can be divided into three categories: modules, visualizations, and interactions. Modules are the fundamental interactive elements that fetch and organize data from the server. Each module renders its own combined visualizations of the content of the messages and conversation metrics. Moderators have several ways to interact with the visualizations and modules to help them navigate through large amounts of data and customize their workspace.

5.1 Modules

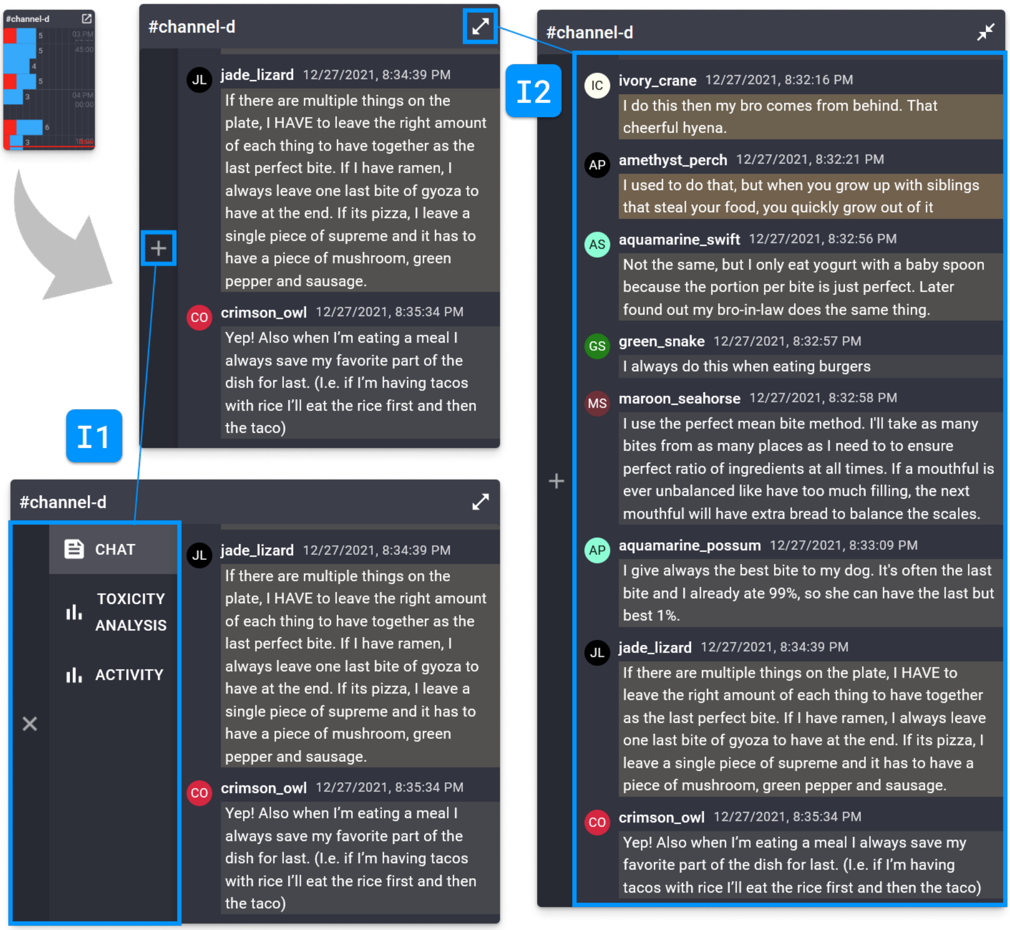
The first module a moderator encounters once a guild is selected is the Guild Overview (Figure 1, M1). This module provides a palette of channel thumbnails which allows the moderator to get a quick overview of the activity in each channel through aggregate statistics. Each channel thumbnail displays a histogram with the total amount of messages sent in blue and the number of messages above a fixed toxicity threshold in red. This design allows moderators to monitor multiple channels at once (**G1**) and helps them decide which channels need a closer inspection based on overall activity and toxicity levels. Additionally, it helps to narrow their focus (**G2**) so they can spend their time investigating potentially problematic exchanges instead of monitoring a channel with mostly benign activity.

The moderator can drag a tile into the workspace area to see a full view of the channel (Figure 1, M2), including the individual messages and toxicity scores. The initial visualization is a message

		Feature	Description	Example Use Case
Navigation Modules	**M1**	Server/Guild Overview	Composed of: *M1.1* Channel Palette for browsing channels *M1.2* Workspace for organizing conversation views (see M2)	Moderator chooses channels to focus on and drags them into the workspace for more comprehensive monitoring
	M2	Conversation View	Displays messages alongside analysis	Moderator reads messages as usual and with the help of analysis to skim through conversations more efficiently
	M3	Channel Tile	Concisely presents preview of the activity and toxicity in a channel	Moderator passively monitors lower risk channels for bursts of unusual or problematic activity
Visualization	**V1**	Heat Map	Uses a sequential color scheme to highlight messages based on scalar values from analysis	Moderator gets a sense of (predicted) toxicity of a conversation while skimming as usual
	V2	Bar Chart	Displays scalar values from analysis as a horizontal bar chart	Moderator utilises exact results of toxicity analysis to find patterns and filter based on analysis (see I5)
	V3	Activity Chart	Displays the volume of messages and the proportion of (predicted) toxic messages sent to a channel within a given period of time	Moderator identifies a channel with a higher volume of activity to focus on
Interactions	**I1**	Interchangeable Charts	Interface for selecting and recording charts to view	Moderator selects relevant charts to display for each channel
	I2	Drag and Drop	Customize the layout of the modules	Moderator arranges the different modules and charts to suit their preferred workflow
	I3	Expand/Collapse	View messages one by one (expanded) with or without text, or within a certain time period (collapsed). Double click on a message to toggle between the expanded and collapsed view centered around that message.	Moderator uses collapsed view to passively monitor a low-risk, fast moving channel, then uses expanded view to investigate conversations in detail
	I4	Hover to Preview	Hover over a bar to preview the message	Moderator reads high toxicity messages directly in the collapsed view
	I5	Toxicity Threshold	Set a threshold of toxicity, and highlight messages/bars above that threshold	Moderator highlights high toxicity messages to focus on

Table 1. Summary of features that are implemented in ConvEx. Each row includes a feature's name, a short description, and an example use case of that feature. Features are divided into three categories: Modules, Visualizations, and Interactions. Modules are high-level components that provide a context for implementing the other features. Visualizations encompass the different presentations of toxicity and activity. Interactions denote the control features that are used to manipulate and navigate the interface.

feed that mimics the layout of the Discord application, augmented with a heat-map visualization of the toxicity analysis. More visualizations can be accessed through the graph drawer (Figure 3, I1). Additionally, the moderator has the option to switch between a compact, minimized view of the channel, and an expanded maximized view that can display more information. Combined with the drag-and-drop interface, this allows the moderator to customize their workspace (**G4**) and get a detailed view of high-priority channels while also having lower-priority channels open on the side.



jade_lizard 12/27/2021, 8:34:39 PM
If there are multiple things on the plate, I HAVE to leave the right amount of each thing to have together as the last perfect bite. If I have ramen, I always leave one last bite of gyoza to have at the end. If its pizza, I leave a single piece of supreme and it has to have a piece of mushroom, green pepper and sausage.

crimson_owl 12/27/2021, 8:35:34 PM
Yep! Also when I'm eating a meal I always save my favorite part of the dish for last. (I.e. if I'm having tacos with rice I'll eat the rice first and then the taco)

jade_lizard 12/27/2021, 8:34:39 PM
If there are multiple things on the plate, I HAVE to leave the right amount of each thing to have together as the last perfect bite. If I have ramen, I always leave one last bite of gyoza to have at the end. If its pizza, I leave a single piece of supreme and it has to have a piece of mushroom, green pepper and sausage.

crimson_owl 12/27/2021, 8:35:34 PM
Yep! Also when I'm eating a meal I always save my favorite part of the dish for last. (I.e. if I'm having tacos with rice I'll eat the rice first and then the taco)

ivory_crane 12/27/2021, 8:32:16 PM
I do this then my bro comes from behind. That cheerful hyena.

amethyst_perch 12/27/2021, 8:32:21 PM
I used to do that, but when you grow up with siblings that steal your food, you quickly grow out of it

aquamarine_swift 12/27/2021, 8:32:56 PM
Not the same, but I only eat yogurt with a baby spoon because the portion per bite is just perfect. Later found out my bro-in-law does the same thing.

green_snake 12/27/2021, 8:32:57 PM
I always do this when eating burgers

maroon_seahorse 12/27/2021, 8:32:58 PM
I use the perfect mean bite method. I'll take as many bites from as many places as I need to to ensure perfect ratio of ingredients at all times. If a mouthful is ever unbalanced like have too much filling, the next mouthful will have extra bread to balance the scales.

aquamarine_possum 12/27/2021, 8:33:09 PM
I give always the best bite to my dog. It's often the last bite and I already ate 99%, so she can have the last but best 1%.

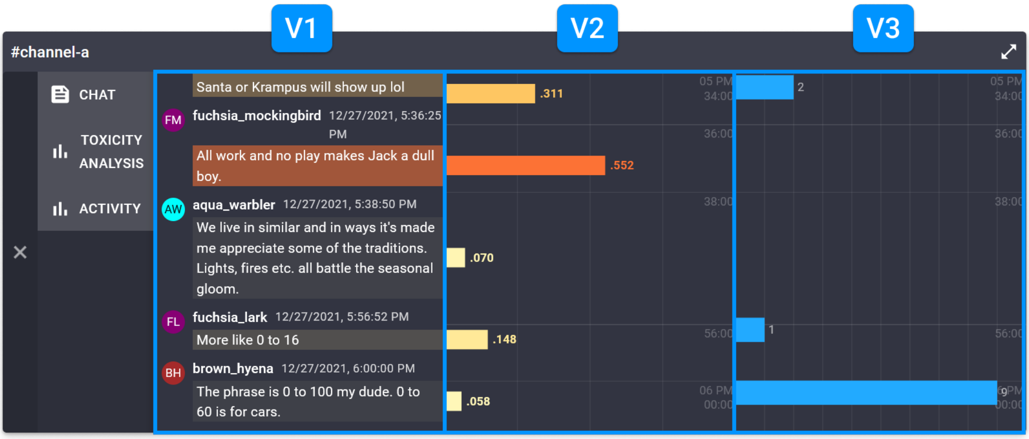
jade_lizard 12/27/2021, 8:34:39 PM
If there are multiple things on the plate, I HAVE to leave the right amount of each thing to have together as the last perfect bite. If I have ramen, I always leave one last bite of gyoza to have at the end. If its pizza, I leave a single piece of supreme and it has to have a piece of mushroom, green pepper and sausage.

crimson_owl 12/27/2021, 8:35:34 PM
Yep! Also when I'm eating a meal I always save my favorite part of the dish for last. (I.e. if I'm having tacos with rice I'll eat the rice first and then the taco)

Fig. 3. Dragging a tile from the channel palette (M3) into an empty area reveals an expanded channel view that displays the content of messages within a channel in more detail. The user can adjust the size of the expanded channel view, toggling between two sizes by clicking the button in the top right corner (B). In the maximized size, the view takes up the whole height of the viewport and is slightly wider than the more compact minimized view, which takes up half the vertical height. This allows the user to mix compact views with larger, more detailed views as in Figure 1.

5.2 Visualizations

Three visualizations of toxicity are offered in the full channel view. Using the graph drawer, the moderator can select which graphs they want to view. These visualizations are designed to help the moderator utilize the toxicity analyses from the Perspective API to help them narrow their focus on potentially problematic conversations (**G2**) and to help them understand the context of a situation faster (**G3**). For example, ConvEx will allow moderators to quickly identify when a heated argument began by looking for when the first "highly toxic" message was posted. This notion of "highly toxic" is likely to correspond to different values for different servers and even different channels within a server. To account for these nuances, we include a configurable, toxicity-based highlighting feature described later in the section.



Santa or Krampus will show up lol

fuchsia_mockingbird 12/27/2021, 5:36:25 PM

All work and no play makes Jack a dull boy.

aqua_warbler 12/27/2021, 5:38:50 PM

We live in similar and in ways it's made me appreciate some of the traditions. Lights, fires etc. all battle the seasonal gloom.

fuchsia_lark 12/27/2021, 5:56:52 PM

More like 0 to 16

brown_hyena 12/27/2021, 6:00:00 PM

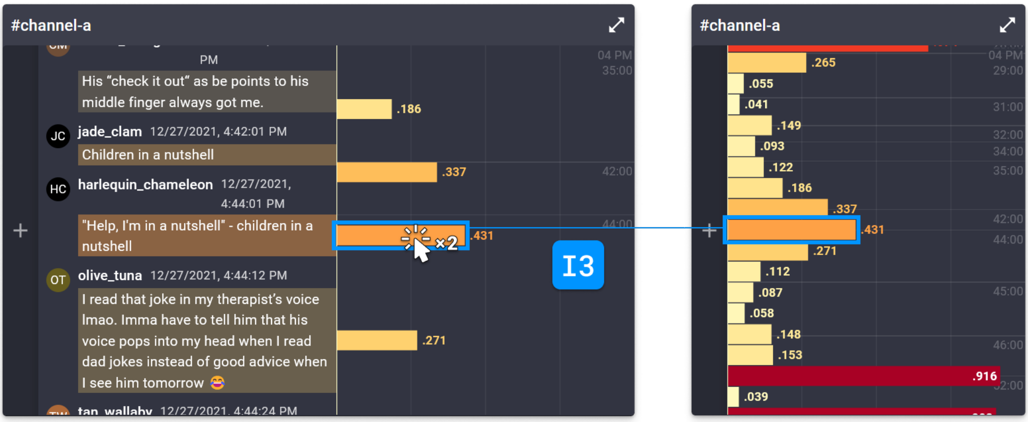
The phrase is 0 to 100 my dude. 0 to 60 is for cars.

Fig. 4. Different visualizations can be accessed by clicking on the "+" button on the left edge of the channel view. This opens a drawer, from which the user can toggle which charts are displayed. In this figure, three visualizations are shown (left to right): heatmap view, bar chart view, and activity view.

The initial view is the heat-map representation of toxicity scores (Figure 4, V1). Messages are displayed to mimic the layout of messages that one would find on the Discord app itself. This familiarity is designed to help keep navigating through messages easy, thus reducing the overhead of adopting a new tool (**G5**). A color is displayed as the background of each message, based on a white-yellow-orange-red sequential scale that corresponds to the predicted toxicity score from the Perspective API. Opacity is also modulated with toxicity score, to make values closer to 1 more apparent. A toxicity probability of 0 corresponds to transparent (white), while a toxicity of 1 corresponds to bright red at full opacity. This helps to give the moderator a rough idea of the toxicity of the message and the surrounding context while reading through the messages as one normally would.

For a more precise reading of the toxicity values, the toxicity analysis is also displayed as a bar chart (Figure 4, V2). The bars are colored according to the same scheme as the heat map and are labeled with the toxicity probability to a precision of 3 digits after the decimal point. In conjunction with the adjustable toxicity threshold interaction described later in this section, the bar chart visualization of toxicity is helpful for making use of the precise toxicity scores. This allows the moderator to take a model-first workflow, utilizing toxicity predictions as a heuristic for conversation health, which is especially important for filtering information while monitoring content on fast-moving servers. The condensed view (Figure 6, I4), allows for faster navigation while retaining the ability to manually inspect individual messages.

Aggregates of analyses are also visualized as the highest level overview of the activity within a channel. The number of messages sent and the number of messages that exceed a fixed threshold of toxicity is counted for each five-minute interval. The total number of messages is represented as a blue bar, and the number of toxic messages is represented as a red bar (Figure 4, V3). Each bar is labeled with the total number of messages sent within the relevant time span. By utilizing aggregates, data about a large number of messages can be visualized in a compact space, making it ideal for the channel thumbnails, as they can provide previews of several channels that can easily be scanned when deciding which channels to view in detail.



PM

His "check it out" as be points to his middle finger always got me.

JC jade_clam 12/27/2021, 4:42:01 PM

Children in a nutshell

HC harlequin_chameleon 12/27/2021, 4:44:01 PM

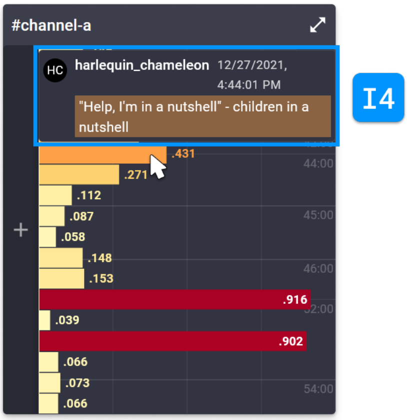
"Help, I'm in a nutshell" - children in a nutshell

OT olive_tuna 12/27/2021, 4:44:12 PM

I read that joke in my therapist's voice lmao. Imma have to tell him that his voice pops into my head when I read dad jokes instead of good advice when I see him tomorrow 😂

tan_wallaby 12/27/2021, 4:44:24 PM

Fig. 5. Different charts may use different scales for the y axis. To make transitions smoother, the moderator can double click on a specific message to initiate an animated transition while keeping the selected message in the frame. This reduces disorientation caused by changing scales.



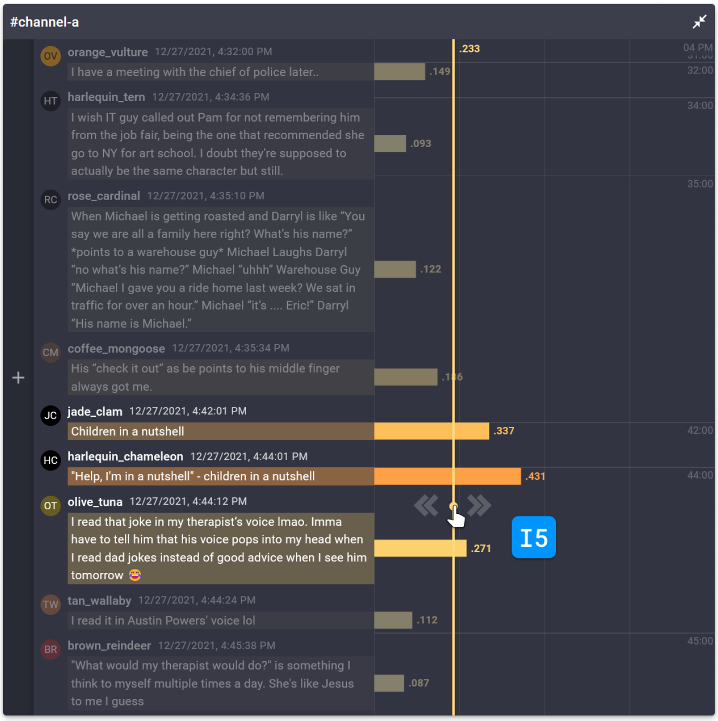
HC harlequin_chameleon 12/27/2021, 4:44:01 PM

"Help, I'm in a nutshell" - children in a nutshell

Fig. 6. When the bar chart view is enabled and the other views are disabled, the layout condenses so that the bars are stacked on top of each other. Contrast this with Figure 4 where the bars are positioned to line up with the corresponding message. From this view, hovering over a bar displays the original message in a tooltip so that the user can quickly access and review the original message corresponding to each bar.

5.3 Interactions

Expanding and collapsing is an implicit interaction that happens when the analysis bars are visible and the heat-map is shown or hidden. When the heatmap is visible, the bars are aligned to each message. As messages can take up an arbitrary amount of height, the bars have an arbitrary distance between them. While this makes sense in the context of the heat-map view which displays the messages, when the bars are isolated, it makes more sense to stack the bars directly above each other. To make this transition without disorientating the viewer, the transition is animated, and the bars in the middle of the viewport remain fixed as the surrounding bars enter or leave the viewport (Figure 5, I3). This helps to ensure that messages from a similar time frame are being shown before



#channel-a

orange_vulture 12/27/2021, 4:32:00 PM
I have a meeting with the chief of police later..

harlequin_tern 12/27/2021, 4:34:36 PM
I wish IT guy called out Pam for not remembering him from the job fair, being the one that recommended she go to NY for art school. I doubt they're supposed to actually be the same character but still.

rose_cardinal 12/27/2021, 4:35:10 PM
When Michael is getting roasted and Darryl is like "You say we are all a family here right? What's his name?" *points to a warehouse guy* Michael Laughs Darryl "no what's his name?" Michael "uhhh" Warehouse Guy "Michael I gave you a ride home last week? We sat in traffic for over an hour." Michael "it's .... Eric!" Darryl "His name is Michael."

coffee_mongoose 12/27/2021, 4:35:34 PM
His "check it out" as be points to his middle finger always got me.

jade_clam 12/27/2021, 4:42:01 PM
Children in a nutshell

harlequin_chameleon 12/27/2021, 4:44:01 PM
"Help, I'm in a nutshell" - children in a nutshell

olive_tuna 12/27/2021, 4:44:12 PM
I read that joke in my therapist's voice lmao. Imma have to tell him that his voice pops into my head when I read dad jokes instead of good advice when I see him tomorrow 😂

tan_wallaby 12/27/2021, 4:44:24 PM
I read it in Austin Powers' voice lol

brown_reindeer 12/27/2021, 4:45:38 PM
"What would my therapist would do?" is something I think to myself multiple times a day. She's like Jesus to me I guess

Fig. 7. ConvEx supports a highlighting feature where the user can visually set a threshold of toxicity probability above which data will be highlighted. This can be used to help direct the user's attention toward important messages.

and after the transition, making it easier for the moderator to re-orient and continue examining the conversation from where they left off.

While the collapsed bar view, the moderator can quickly preview the full message content by hovering over one of the bars (Figure 6, I4). This allows the moderator to manually inspect any messages that may have been flagged with a high toxicity score, and quickly determine whether or not it needs further attention. The moderator can also highlight messages and analyses that exceed a certain toxicity threshold, to filter for relevant information (**G2**). The threshold can be adjusted by using the draggable ruler included in the bar-chart (Figure 7, I5), which provides live visual feedback and allows the moderator to use existing messages as reference for where to set the threshold.

Additional features were implemented to ensure that ConvEx can be customized to fit the specific needs of a moderator and their community (**G4**). The moderator has the ability to select which charts are shown by expanding the graph drawer and can select different charts for each channel. The drag and drop interface combined with the maximization/minimization feature allows the moderator to customize their workspace and arrange the visualizations in a way that makes sense for their workflow and their community (Figure 1, M1.2).

6 EVALUATION

We conducted an online, synchronous user study with moderators of various Discord servers. Having a lack of responses after messaging moderators directly on Discord, we opted to find subreddit communities of appropriate size that had an associated Discord server, then ask them via modmail to refer us to their Discord moderators. Through this process, we recruited a total of eight participants, all from different servers with an average size of about 38,000 members (see Table 2). Participants were required to have at least 3 months of experience moderating a server with at least one thousand members and were compensated with a $20 Amazon gift card.

The study, which took ~1 hour for each participant, began with a brief introduction to the features of ConvEx (see Table 1). Participants were then led through a series of tasks to simulate the use of ConvEx in an active moderation setting while a researcher made observations on which features were used and how. This was accompanied by a semi-structured interview where participants were asked about their experiences as moderators on Discord and were also asked to provide their own responses to the features and design elements of ConvEx. The observations and transcripts from the studies were independently coded by two of the authors to: a) get insights into how moderators interpret and utilize the various approaches to integrating AI into the workflow implemented in ConvEx, b) understand how the different components within the interfaces of ConvEx and Discord affect how moderators approach their tasks, c) identify which design elements and features the moderators valued as a potential addition to their workflow, and d) get feedback on the overall design and implementation of ConvEx for future iteration.

6.1 Simulated Server for User-Testing

We designed our testing environment to replicate typical moderation conditions in an effort to ensure ecological validity. Ideally, we would have liked to deploy and evaluate ConvEx on a real Discord server. In order to fetch messages from a Discord server and display them in ConvEx, a third-party bot had to be present on that server and given appropriate permissions. However, most participants were hesitant to add a third-party bot to their server. As an alternative, we created a simulated environment that mimicked the following aspects of a Discord server—conversation "threads", activity distributed across multiple channels, and timing between messages. The content for this simulated server was sourced from comments on top Reddit submissions fetched using PRAW[4], which were then organized into sequences such that each sequence of comments would resemble a conversation thread (see Appendix A). Our simulated server consisted of 8 different channels. Each channel in the simulated server would randomly choose a sequence and push each comment from the sequence to the chat window as if it were a regular Discord message. The timing between messages was made to be 60 times faster than the original time between comments (which was derived from the original timestamps of the corresponding comment on Reddit), i.e., two comments posted one minute apart on Reddit would be sent one second apart in the simulated server. The content of this simulated server was pre-built, then "replayed" on demand for each study, with a delay of about 2 to 20 seconds between messages (see Appendix A for specific details). For this study, 206 sequences with a mean length of 20.77 messages were collected.

6.2 Evaluation Tasks

Following an initial interview and a brief walkthrough of the features in Table 1, the participant was asked to attempt the following tasks using ConvEx on the simulated server.

(1) Choose a channel and try to find one or two events of potential interest in the past. To the best of your ability, try to verbalize how you are using each feature in your search process.

[4] https://praw.readthedocs.io/en/stable/

Participant	Approx. Server Size	Approx. Time Moderating on Discord
P1	7,000	1 year
P2	132,000	1.5 years
P3	116,000	1 year
P4	200	2 years
P5	1,600	4 years
P6	16,000	10 months
P7	27,000	4 years
P8	3,000	1.5 years

Table 2. Table of participants in the user study. Participants were recruited by messaging the modmails of subreddits that had an associated Discord server. Participants were required to have at least 3 months of experience moderating on Discord. The mean server size our participants moderated for was about 38,000 members, and the mean time as a Discord moderator was about 2 years.

(2) Try to determine the context of the situation, who was involved, and actions you might take.
(3) Return to the guild overview and try to identify which channel(s) warrants the most attention at the moment. Tell us how you are making that decision.

These tasks were based on responsibilities that moderators often take on, including investigating reports from the community, catching up on the circumstances of an incident, and actively monitoring a server, and were designed to simulate a condensed version of a moderator's regular duties, focusing on the challenges we identified in Section 3. We aimed to identify the features that were most or least effective in helping the moderators in both retroactive moderation (e.g., catching up on conversations) and proactive moderation (e.g., keeping up with a fast-moving server and deciding which channels to actively monitor).

7 FINDINGS

Overall, our moderators indicated several ways in which ConvEx would support them in their day-to-day moderation duties. Six moderators (P1, P2, P4, P6, P7, and P8) felt that ConvEx best supported their ability to locate problematic messages. P2 and P3 indicated that they felt that ConvEx could help them work faster and P8 specifically indicated that ConvEx would reduce their cognitive load. P1, P2, P3, and P4 also indicated that ConvEx would be of significant help during periods of high activity within the server, especially when activity is spread across several channels. While most moderators had expressed positive overall impressions, some concerns included the cost of learning and maintaining a new tool and the low accuracy of the underlying model, which limits the benefit ConvEx can offer, especially for servers with relatively low activity. Overall, 7 of the 8 moderators expressed their confidence that ConvEx would be a valuable addition to their workflow, complementary to their existing toolkit.

In the following subsections, we describe specific findings of how moderators used each of ConvEx's features, how valuable each feature was to the moderator, and which features moderators found most valuable in their workflow. We also detail how ConvEx supported specific moderation strategies for various workflows in which moderators actively read through or passively observe their server for areas that need attention. We document the think-aloud responses and specific feedback from moderators, as well as our own observations of their interaction with ConvEx during

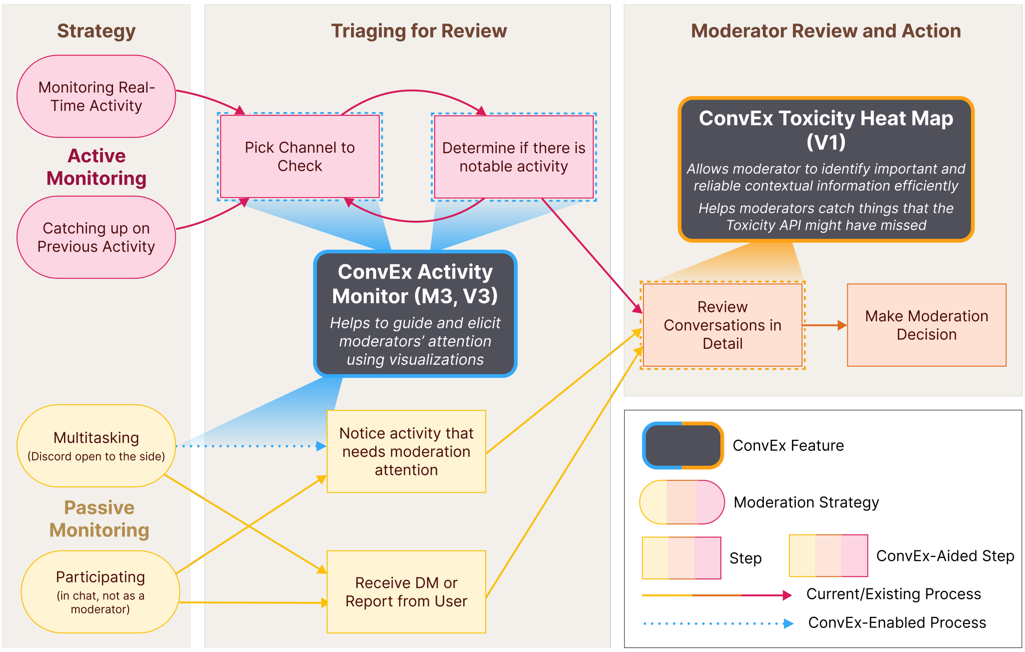

Strategy
Triaging for Review
Moderator Review and Action
Monitoring Real-Time Activity
Active Monitoring
Catching up on Previous Activity
Pick Channel to Check
Determine if there is notable activity
ConvEx Activity Monitor (M3, V3)
Helps to guide and elicit moderators' attention using visualizations
ConvEx Toxicity Heat Map (V1)
Allows moderator to identify important and reliable contextual information efficiently
Helps moderators catch things that the Toxicity API might have missed
Review Conversations in Detail
Make Moderation Decision
Multitasking (Discord open to the side)
Passive Monitoring
Participating (in chat, not as a moderator)
Notice activity that needs moderation attention
Receive DM or Report from User
ConvEx Feature
Moderation Strategy
Step
ConvEx-Aided Step
Current/Existing Process
ConvEx-Enabled Process

Fig. 8. In this figure, we depict the most common processes employed by the moderators we interviewed to monitor and review the content on their servers. We highlight four main strategies (rounded boxes), representing the intermediate steps with the square boxes, and representing the transitions between steps in their current process with the solid arrows. We also highlight two of the features that moderators found the most valuable: the activity overviews (blue-outlined box) and the heatmap (orange-outlined box). The highlighted regions indicate which step in the process moderators found the feature most helpful for. While moderators are actively monitoring, the activity overviews help moderators in picking which channel to check and determining if there is any notable activity by guiding their attention to potentially problematic areas of the server. The heatmap helps moderators in reviewing content in detail by making it easier to obtain reliable information from the AI generated cues while reading through each message. When moderators are passively monitoring, the activity overviews elicit moderators attention when potential problems occur, enabling a new workflow represented by the dotted blue arrow, where moderators can directly notice activity that needs their attention even when engaged in other tasks without having to rely on reports or DMs from users.

the study. Using these data, we analyze how moderators interpreted the information presented through the ConvEx interface, and how ConvEx supports their moderation strategies and workflow. Additionally, we illustrate the most common workflows and how ConvEx supports them in Figure 8.

7.1 Impressions of Individual Features

We present a breakdown of ConvEx's features to analyze which aspects moderators found to be most valuable, and the role that each navigation feature, visualization, and interaction played in supporting their workflow. Overall, we found that moderators found ConvEx most helpful for locating and identifying specific instances of problematic behavior as well as for finding concrete evidence thereof. The most important features were the activity overviews in the channel palette (V3, M3) for identifying channels to narrow their focus on, and the heatmap visualization for locating and analyzing parts of a conversation that need careful attention.

Navigation Modules. We observed several common navigation patterns that moderators would employ when moderating with ConvEx. In the most common pattern, moderators proactively monitored the activity overviews (V3) embedded in the channel tiles (M3) in the channel palette (M1.1) to locate the channel with the highest levels of toxicity and/or activity, then expand it to a full chat view (M2) to read through the messages within in the channel. 6 of 8 moderators took this approach from the start of the study. The remaining two moderators (P5, P6) used the activity overview (V3, M3) at the start to narrow down a fixed set of highly active "main" channels to pay attention to. These moderators had reported that they primarily relied on their knowledge of which channels tended to be the most problematic (e.g., a politics channel) when moderating in their own server, but they quickly identified the most active channels during the study by utilizing the activity overviews, suggesting that the overviews may also play a role in helping new moderators adapt to their servers. Beyond this initial narrowing phase, the activity overviews (V3, M3) were not utilized as frequently, as these moderators spent most of their time reading through the messages in the channel view (M2) they had open, though they switched focus when they noticed a spike of activity in another channel they previously identified through the activity overviews (V3, M3).

Visualizations. Within the expanded channel view (M2), all moderators had used the heatmap chart (V1) while reading through messages. P6 and P7 said they particularly valued the toxicity visualizations in the heatmap and the bar chart (V2) for their ability to attract their attention to problematic areas. P3 and P5 specifically pointed out that the bar charts (V2) had been helpful in locating potentially toxic messages. P2 had made frequent use of the bar chart to determine where to set the toxicity threshold (I5) for highlighting certain messages. However, P6 felt the bar chart was redundant as the exact value of toxicity was not particularly important and the color scale on the heatmap was sufficient.

As discussed above, the activity/toxicity charts (V3) within the channel tiles (M3) were very helpful for deciding where to focus their attention. Several moderators (P1, P2, P3, P4, P6, P7) had also remarked they were useful for getting a quick sense of the overall health of a server, especially when trying to manage a large server with many channels:

> *"I liked the breakout on the side, to see activity levels and just a quick vibe check. I don't really like that phrase, but I guess that's what people say nowadays, vibe checks of conversations." -P1*

However none of the moderators made active use of the activity chart (V3) within the expanded view (M2), some reporting that it was more difficult to interpret or that it provided little additional information in the latter context (P1).

Interactions. We observed a few patterns in how moderators would use the layout interactions (I2) to organize their workspace (M1.2). Some moderators seemed to prefer focusing on only one or two channels at once, minimizing each channel view (M2) back to the palette (M1.1) after they were done reading through. This, perhaps, mimicked a familiar workflow on Discord where only one channel can be viewed at a time. Others preferred to keep several of the most active channels open to monitor them live and stay up-to-date, a feature that several moderators (P1, P3, P5) found particularly valuable as Discord currently only supports viewing one channel at a time. While most moderators did not specifically comment on the size of the charts, P2 expressed it would be preferable to have just one expanded channel at a time at full size:

> *"I'll just be having [the channel view] in expanded [mode] all day because I've been going through channel to channel anyway, so it makes sense for me to just have the whole thing open, finish one channel and then get back to another channel." -P2*

This suggests that it would be valuable to support at least two distinct layouts: a "multifaceted mode" that allows for many channels to open at once (which we have implemented), and a separate "focus mode" that more closely mimics the one-channel-at-a-time layout of Discord.

Moderators routinely made use of I1 to choose the combination of visualizations that worked for them, which tended to vary as moderators expressed preference for different combinations of visualizations. Although moderators did not specifically comment on interactions I3 and I4, we observed that they were most often encountered when moderators were interpreting particularly high toxicity scores, as I4 offered a quick preview of the message content and I3 revealing the context around it. One moderator (P2) found the toxicity threshold slider (I5) particularly helpful for highlighting the messages they would want to review, and saw potential in using the threshold to isolate those messages into a separate view. However, most moderators did not utilize or comment on the feature, likely due to an oversight in our interface design, as it requires the bar chart view to be enabled first, and it is easy to miss if one does not already know it exists. Nonetheless, we found that it is an important feature as it allows ConvEx to serve different needs regarding precision. While one moderator (P5) stated that low precision (high false-positive rate) limits the usefulness of ConvEx, another (P2) stated they prefer the higher recall even at the expense of low precision. Though only the latter made use of the feature during the study, both agreed that an adjustable threshold to refine the toxicity predictions is a desirable feature.

7.2 Impact on Active Moderation Strategies

We present our observations on how ConvEx supports active moderation strategies where the moderator is actively trying to keep up with the activity in their server, or when they are reading the history of messages to find any incidents they missed. We analyze the aspects of ConvEx in the framework of creating a synergy between automated techniques and human judgement, where automated techniques provide cues that help the human decide where to focus while the human catches nuances that automated techniques cannot. We focus on how the different aspects of our design guided moderators' attention to important areas, as well as how our design helped moderators extract reliable information and catch things that the automatic analysis did not.

Guiding Attention through Visualization. Guiding moderators through the analysis of a conversation has the potential to help moderators analyze and respond to such behavior more quickly. By giving moderators cues to quickly skim over low-risk parts and focus their attention on high-risk areas instead, we can reduce their mental load. Our findings suggest that the activity overviews (V3, M3) were valuable for guiding attention at a high-level, as several (P1, P2, P3, P4, P6, P7) found them helpful for deciding where to focus their attention, especially in servers with many of channels.

> *"Because with 20, 30, 40, 50 channels on a server, scrolling through this sidebar to see where in the heat maps something is marked red is much easier than going through every single channel, clicking through it, scrolling through all the messages I missed." -P3*

Our observations also suggest that, for guiding attention during more fine-grained analysis, the heat map overlaid on the original text, on its own or in conjunction with the bar chart, can be an effective way to guide a moderator's attention toward potentially problematic behavior while retaining the original text and the nuances within.

> *"The heatmaps ... gave me an indication perhaps of which comments ... to read in more detail and not skim through." - P7*

Several also (P6, P7) remarked that they were naturally drawn to read messages with a high (predicted) toxicity score more carefully. P2 felt it was helpful for identifying the most problematic messages consisting an incident.

> *"This makes it easy for me to find the offending message and get rid of it quickly" - P2*

However, we also note that some moderators found that it was somewhat more difficult to track messages sent by the same user, making it difficult to analyze the user-level context in a conversation.

> *"I think with these different colors, it makes it a little hard to sort out by user. Because a lot of times it's like specific users that are being the most offensive or the most aggressive" - P1*

This highlights the importance of considering what kind of patterns in the data a particular visualization highlights, and what kind of patterns it obscures. In Section 8.3, we discuss future additions to ConvEx that aim to elucidate patterns within a user's history.

Helping Moderators Obtain Reliable Information. We investigate how design choices in the interface affect moderators' confidence in the model and their ability to correctly interpret and synthesize trustworthy information from its predictions, since this affects their willingness to offload some of the work of analysis to a computer. Overall, the perceived accuracy of the predicted toxicity scores as presented in ConvEx varied, and would even change throughout a single session. While a few moderators (P1, P2, P5, P7) had felt the accuracy of the toxicity model was less than ideal and appeared to produce a noticeable amount of false-positives, most also pointed out that they felt the model was still accurate enough to reliably convey a general impression of a conversation's health (P1, P2, P4, P6, P7).

We observed from P2 that customizing parameters such as the toxicity threshold (I5) can increase the moderators' confidence in their interpretation of the predictions. We also found, that moderators (P4, P6, P7) expressed a greater degree of confidence when interpreting the predicted toxicity scores for a cluster of consecutive, high-scoring messages than for an isolated, high-scoring message. The heatmap and bar chart made visual identification of those clusters possible, while the activity overview directly presented aggregate information to help locate those clusters. We find that making such clusters easier to locate and identify makes it easier for our moderators to confidently make inferences from automated analysis.

Helping Moderators Catch Things the Algorithm Missed. The visualizations help moderators to visually filter out messages and decide which ones to read carefully or to skim over, thus saving them work while highlighting messages that might need more careful attention. However, by design, moderators are also able to see messages that were not rated as highly-toxic, keeping the context surrounded the messages intact and allowing for moderators to catch nuances that the algorithm missed. In several instances, moderators were able to identify messages and users they would have sanctioned (but which were not highlighted in the interface) by spotting some red flags such as all caps or political topics while skimming through the messages.

We also found that, when carefully reading through and analyzing messages, several moderators (P2, P5, P7) attempted to reason about why the underlying algorithm had given the message the toxicity score it did, then construct on argument based on their mental model of the algorithm. In particular, when moderators believed the model made an error (i.e., a message was scored too low or too high), they would first identify specific words (e.g., swears) or tone (e.g., anger) that they believed the model had picked up on. They would also identify cues from the surrounding context and draw from their personal moderation experience to explain why they thought the model's output was incorrect.

7.3 Impact on Passive Moderation Strategies

We found that Convex has the capacity to support passive moderation strategies, where moderators can keep an eye on the server without taking their attention away from other duties. P3, P5, and P7 specifically mentioned that ConvEx allows them to focus on one channel while also keeping an eye on the other channels, which they expect would be especially helpful when moderating a Discord server with several active channels.

> *"I'd probably have some of the most active channels here at once. Move the window to my second screen, work on something else. Moderating is often not a too active thing, and [I'd] passively [watch] the messages coming in on the sides." - P3*

Several moderators (P4, P8) also had indicated their team consists mainly of volunteers, and ConvEx could be especially valuable for fewer missed responses and more timely interventions, especially when moderator capacity is low, such as on holidays.

> *"This would be very helpful in a situation where, for whatever reason, there's just not a lot of moderators active at that time. Actually, we had this a bit around Christmas, because there's obviously a lot of people who are busy, off with their families, et cetera. This would allow a single moderator, a few moderators to keep an eye, basically, on what's going on across the server." - P7*

8 DISCUSSION

ConvEx navigates the complex tradeoffs present when building an AI-augmented, mixed-initiative system [24]. Specifically, we consider three key tradeoffs. First is the tradeoff between accuracy and precision, where creating a more precise AI trades off in accuracy and vice versa. Next is the tradeoff between automation and agency. While a highly automated system is fast, giving the user the agency to make the final decision mitigates the effects of an erroneous prediction from the AI. The final tradeoff is between configurability and usability. A highly configurable system can theoretically be tweaked to adapt to many different situations. However, more configurability necessarily increases the complexity of the user interface.

With ConvEx, we balance these tradeoffs by communicating AI predictions in a way that leaves room for moderator interpretation. In effect, we pass the initiative back to the user to decide how much attention to spend on a particular message based on their experience and capabilities. We reduce the amount of explicit configuration required beforehand for different contexts, instead leveraging a human moderator's ability to adapt implicitly (G4). The visualization augments moderators' existing strategies when reading through and analyzing messages, including looking for harsh words, tone, or previously problematic users by adding signals they can use to quickly assess a situation (G2, G3). By focusing on designing an interface that augments moderators' existing strategies, we are able to design a system that was intuitive for moderators to use and complementary in their workflow (G5).

Next, we share implications for the design of future AI-augmented systems for moderation, augmenting the interface with conversation metrics while leaving room for interpretation, and supporting varying levels of granularity in moderation tasks. Finally, we also discuss high-impact extensions to our current design and implementation of ConvEx, including other conversational metrics (in addition to toxicity and activity levels) that would help moderators proactively identify and respond to problematic behavior.

8.1 Designing for Flexible Interpretation

Signals generated from computed conversation metrics help guide moderators on what they need to focus on. Examples include the usage of strong words, all caps, and keywords related to controversial topics. Computed signals—e.g., synthesized social signals (S3s) in a system like Sig [27]—have the capability to summarize large amounts of data much faster than a human can, then display these results to guide users toward certain profiles. These computed signals can be discrete or continuous.

In the case of a discrete signal (e.g., toxicity markers in Sig), their accuracy is limited by the fact that they are either correct or incorrect. For example, Sig marks Twitter profiles as likely toxic or not, which leaves little room for interpretation, and if taken at face value, can be unfair or inaccurate. However, continuous signals (e.g. star ratings, list rankings) are more forgiving, as they allow for a wider range of interpretation and thus widen the margin of error. For example, on Google Maps, a star rating for a restaurant of 4.0 stars does not necessarily indicate a good or bad restaurant and requires the person making the decision to decide whether 4.0 stars is high enough to be worth the risk, or if they want to investigate further. Even though the AI prediction does not lead directly to a decision that can be automated, it helps the user by summarizing and translating relevant information into a simple metric they are familiar with.

For ConvEx, where the application is moderation, the toxicity score is represented as a real value between 0 and 1 giving the moderator more freedom in how to interpret the model's output. The boundary between a "low" score and a "high" score is usually not as simple as a naïve threshold, but depends on the context of the message, in addition to the norms within the channel and server. Because there are many factors involved in making such a decision, and some that might only be implicitly understood, it is easier for a human to implicitly adapt to the nuances (which algorithms will miss) between different communities and contexts. Therefore, we designed the visualizations to guide moderators towards areas of concern, and allow them to use their experience and knowledge to make judgements (G2). Visual features, such as brighter colors for higher toxicity scores, or summations of data, such as the bar charts, were designed to take advantage of a moderator's visual intuition. Though the visualizations may lack precision, they still retained enough information to be useful to moderators. For designing similar systems in the future, we propose that communicating AI predictions in a manner that allows for a flexible interpretation works with the innate ability of people to adapt, thus suiting the needs for a broader range of communities and moderators (G4).

8.2 Supporting Different Levels of Moderator Oversight

It is important for a moderation system to provide useful information at different levels of granularity, and thereby allow moderators to do their jobs in a top-down approach, with different levels of oversight and focus. This is critical as moderators are often unpaid volunteers with a finite amount of time and attention they can dedicate to moderating, especially as they are outnumbered by users and active conversations and are often unable to keep up with the pace of conversation. With the current interface of Discord, it is difficult for moderators to monitor multiple conversations live or take action on one channel without taking their attention away from all the others. Currently, moderators often rely on user reports (e.g. pinging/tagging) to notify them of where their attention is needed.

ConvEx enables Discord moderators to passively monitor several channels within the server, narrow their attention to the most important channels, and locate problematic conversations and messages within a channel quickly. The activity overview, which provides high-level summaries of behavior that are quickly interpreted, enables moderators to passively observe the overall health and activity of the server while performing other tasks, including monitoring other channels or servers (G1). Within a specific channel, while single-point analyses of messages yield useful information (e.g., toxicity), discourse-level metrics that analyze multiple messages are useful for forecasting conversational outcomes. In ConvEx, we employ the strategy of computing discourse-level metrics by aggregating single-point metrics of several messages within a given time span (see feature V3 in Figure 4), highlighting the trajectory of a conversation.

Moderators were also given the option to view single-point analyses, such as through the heatmap, allowing for a finer grained reading. The heatmap also allows moderators to read through messages sequentially, enabling them to use familiar strategies while they learn and adopt the tool incrementally (G5). We found that the visual grouping of related signals, e.g., multiple messages in a row with a high toxicity score on the heatmap or multiple periods with high toxicity in the activity chart, was interpreted as a single aggregate (in addition to individual scores) with greater perceived accuracy compared to individual scores. This implicit aggregation and synthesis of low-level (e.g., message or sentence level) signals represent a strategy for aiding users in higher-level (e.g., document-level) synthesis. We claim that one strategy for highlighting the trajectory of a conversation (G3) is to synthesize and present users with discourse-level metrics, from aggregation of single-point metrics, such as in the activity chart. Another would be to present single-point metrics in series, as in the heatmap and barchart.

8.3 Next Steps for ConvEx

We plan to extend ConvEx in the following ways:

Visualizing User History. In a future version of ConvEx, our priority would be to add features that facilitate analyzing patterns of user activity. Many of our participants usually take into account the history of the users involved before making a judgment or a decision. However, our findings indicated that the interface may be introducing some friction when moderators are trying to isolate user-level patterns. Following our participants' suggestions, we would like to implement separate views that isolate a user's past messages and allow moderators to add flags to certain users that would be rendered next to their name or icon in ConvEx for keeping track.

Collecting Moderator Feedback to Refine AI. We would also like to include a feature that allows moderators to correct the toxicity score of messages. Considering moderators as domain experts, their feedback can be valuable sources of ground truth. For example, moderators could flag messages as toxic or dismiss ones that are not, which can be used to further refine the underlying model. Further, as we found during the user study, moderators are able to point out what part of the message they believe led to an erroneous prediction. This kind of detailed feedback could be suitable for training attention-based toxicity models as well.

Facilitating the Review of User-Submitted Reports. As another possible extension, we would involve more features that facilitate the review of reports submitted by members of the community. For example, although one participant believed that ConvEx would be helpful in investigating user-submitted reports, the overall impression from participants was that ConvEx was not necessary for this use case. In particular, participants indicated that they would not rely on the toxicity scores when reading through the context of a user-submitted report, since they already identified that the content of the report is important. This suggests that the moderators value human judgment over the algorithmic toxicity scores and that once a problematic event is identified with reasonable certainty, they would prefer to read through the entire context surrounding the event when deciding what further action to take. This process could be augmented with a mod queue (similar to Reddit's own [10]) that links to the original messages in the channel in addition to the in-line user summaries proposed above.

Incorporating Additional Conversational Metrics. ConvEx was designed to easily support other conversational metrics with only slight modification to substitute the Perspective API in the analysis pipeline (see Section 4.2 and Figure 2) with a different API or algorithm. Thus ConvEx can be extended to integrate models that forecast conversational trajectories [38], allowing for early intervention before a conversation breaks down [61]. A model that predicts controversiality [22] would also be helpful in providing early warning signs for moderators. Prosocial metrics [5] can also be integrated to support moderation techniques that focus on promoting positive behavior to build a healthier community. Even non-text models, such as those that use images to detect cyberbullying [25, 62], can be used, as messages in chat are not limited to text, often containing images, videos, links, or emojis. And as new computational models are developed, ConvEx can be extended to keep up with the latest advances in AI-assisted moderation.

This sort of extensibility will be paramount in the future of moderation systems as online communities and the technologies around them continue to evolve at a rapid pace [4]. Moderators, who are the end users of these systems, and software developers, often third-party individuals unaffiliated with a particular platform [34], will need to continuously innovate and adapt to changing circumstances. A framework or standards for these unofficial, community developed moderation systems may become necessary to facilitate the development, deployment, distribution, and adoption of such tools like ConvEx, Sig [27], and others in the future.

9 CONCLUSION

With ConvEx, we present a system to bridge the gap between recent advances in AI research and moderation practices. We designed an interface that specializes in augmenting unstructured chat with visualizations of conversational metrics that aid moderators in monitoring several conversations simultaneously. This allows moderators to quickly locate conversations with problematic trajectories and take proactive action. ConvEx can be easily extended to incorporate other conversational metrics, including those derived from non-text features. Our mixed-initiative interaction design enables flexible interpretation of AI predictions, allowing moderators to implicitly incorporate their own contextual knowledge when interpreting the AI output. We hope that our findings will inform future approaches to creating AI-augmented systems for moderators.

REFERENCES

[1] [n. d.]. Discord Moderator Academy. https://discord.com/moderation

[2] [n. d.]. Perspective API. https://www.perspectiveapi.com/

[3] Shiza Ali, Mohammad Hammas Saeed, Esraa Aldreabi, Jeremy Blackburn, Emiliano De Cristofaro, Savvas Zannettou, and Gianluca Stringhini. 2021. Understanding the effect of deplatforming on social networks. In *13th ACM Web Science Conference 2021*. 187–195.

[4] Tanvi Bajpai, Drshika Asher, Anwesa Goswami, and Eshwar Chandrasekharan. 2022. Harmonizing the Cacophony with MIC: An Affordance-aware Framework for Platform Moderation. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (2022), 1–22.

[5] Jiajun Bao, Junjie Wu, Yiming Zhang, Eshwar Chandrasekharan, and David Jurgens. 2021. Conversations Gone Alright: Quantifying and Predicting Prosocial Outcomes in Online Conversations. *arXiv preprint arXiv:2102.08368* (2021).

[6] Jie Cai and Donghee Yvette Wohn. 2019. Categorizing live streaming moderation tools: An analysis of twitch. *International Journal of Interactive Communication Systems and Technologies (IJICST)* 9, 2 (2019), 36–50.

[7] Jie Cai, Donghee Yvette Wohn, and Mashael Almoqbel. 2021. Moderation visibility: Mapping the strategies of volunteer moderators in live streaming micro communities. In *ACM International Conference on Interactive Media Experiences*. 61–72.

[8] Samuel Carton, Qiaozhu Mei, and Paul Resnick. 2020. Feature-Based Explanations Don't Help People Detect Misclassifications of Online Toxicity. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 14. 95–106.

[9] Stevie Chancellor, Jessica Annette Pater, Trustin Clear, Eric Gilbert, and Munmun De Choudhury. 2016. # thyghgapp: Instagram content moderation and lexical variation in pro-eating disorder communities. In *Proceedings of the 19th ACM conference on computer-supported cooperative work & social computing*. 1201–1213.

[10] Eshwar Chandrasekharan, Chaitrali Gandhi, Matthew Wortley Mustelier, and Eric Gilbert. 2019. Crossmod: A cross-community learning-based system to assist reddit moderators. *Proceedings of the ACM on human-computer interaction* 3, CSCW (2019), 1–30.

[11] Eshwar Chandrasekharan, Shagun Jhaver, Amy Bruckman, and Eric Gilbert. 2022. Quarantined! Examining the effects of a community-wide moderation intervention on Reddit. *ACM Transactions on Computer-Human Interaction (TOCHI)* 29, 4 (2022), 1–26.

[12] Eshwar Chandrasekharan, Umashanthi Pavalanathan, Anirudh Srinivasan, Adam Glynn, Jacob Eisenstein, and Eric Gilbert. 2017. You can't stay here: The efficacy of reddit's 2015 ban examined through hate speech. *Proceedings of the ACM on Human-Computer Interaction* 1, CSCW (2017), 1–22.

[13] Eshwar Chandrasekharan, Mattia Samory, Shagun Jhaver, Hunter Charvat, Amy Bruckman, Cliff Lampe, Jacob Eisenstein, and Eric Gilbert. 2018. The internet's hidden rules: An empirical study of Reddit norm violations at micro, meso, and macro Scales. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–25.

[14] Justin Cheng, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec. 2014. How community feedback shapes user behavior. In *Eighth International AAAI Conference on Weblogs and Social Media.*

[15] Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A computational approach to politeness with application to social factors. *arXiv preprint arXiv:1306.6078* (2013).

[16] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, Vol. 11. 512–515.

[17] Maeve Duggan. 2020. Online harassment. https://www.pewresearch.org/internet/2014/10/22/online-harassment/

[18] Eric Gilbert. 2013. Widespread underprovision on reddit. In *Proceedings of the 2013 conference on Computer supported cooperative work*. 803–808.

[19] Sarah A Gilbert. 2020. " I run the world's largest historical outreach project and it's on a cesspool of a website." Moderating a Public Scholarship Site on Reddit: A Case Study of r/AskHistorians. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW1 (2020), 1–27.

[20] Tarleton Gillespie. 2018. *Custodians of the Internet: Platforms, content moderation, and the hidden decisions that shape social media.* Yale University Press.

[21] Hussam Habib, Maaz Bin Musa, Muhammad Fareed Zaffar, and Rishab Nithyanand. 2022. Are Proactive Interventions for Reddit Communities Feasible?. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 16. 264–274.

[22] Jack Hessel and Lillian Lee. 2019. Something's Brewing! Early Prediction of Controversy-causing Posts from Discussion Features. *arXiv preprint arXiv:1904.07372* (2019).

[23] Manoel Horta Ribeiro, Shagun Jhaver, Savvas Zannettou, Jeremy Blackburn, Gianluca Stringhini, Emiliano De Cristofaro, and Robert West. 2021. Do platform migrations compromise content moderation? evidence from r/the_donald and r/incels. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–24.

[24] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 159–166.

[25] Homa Hosseinmardi, Sabrina Arredondo Mattson, Rahat Ibn Rafiq, Richard Han, Qin Lv, and Shivakant Mishra. 2015. Detection of cyberbullying incidents on the instagram social network. *arXiv preprint arXiv:1503.03909* (2015).

[26] Jane Im, Eshwar Chandrasekharan, Jackson Sargent, Paige Lighthammer, Taylor Denby, Ankit Bhargava, Libby Hemphill, David Jurgens, and Eric Gilbert. 2020. Still out there: Modeling and identifying russian troll accounts on twitter. In *12th ACM Conference on Web Science*. 1–10.

[27] Jane Im, Sonali Tandon, Eshwar Chandrasekharan, Taylor Denby, and Eric Gilbert. 2020. Synthesized social signals: Computationally-derived social signals from account histories. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[28] Shagun Jhaver, Iris Birman, Eric Gilbert, and Amy Bruckman. 2019. Human-machine collaboration for content regulation: The case of Reddit Automoderator. *ACM Transactions on Computer-Human Interaction (TOCHI)* 26, 5 (2019), 1–35.

[29] Shagun Jhaver, Amy Bruckman, and Eric Gilbert. 2019. Does transparency in moderation really matter? User behavior after content removal explanations on reddit. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–27.

[30] Shagun Jhaver, Quan Ze Chen, Detlef Knauss, and Amy X Zhang. 2022. Designing Word Filter Tools for Creator-led Comment Moderation. In *CHI Conference on Human Factors in Computing Systems*. 1–21.

[31] Shagun Jhaver, Sucheta Ghoshal, Amy Bruckman, and Eric Gilbert. 2018. Online harassment and content moderation: The case of blocklists. *ACM Transactions on Computer-Human Interaction (TOCHI)* 25, 2 (2018), 1–33.

[32] Jialun Aaron Jiang, Charles Kiene, Skyler Middler, Jed R Brubaker, and Casey Fiesler. 2019. Moderation challenges in voice-based online communities on discord. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–23.

[33] David Jurgens, Libby Hemphill, and Eshwar Chandrasekharan. 2019. A Just and Comprehensive Strategy for Using NLP to Address Online Abuse. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 3658–3666. https://doi.org/10.18653/v1/P19-1357

[34] Charles Kiene, Jialun Aaron Jiang, and Benjamin Mako Hill. 2019. Technological frames and user innovation: exploring technological change in community moderation teams. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–23.

[35] Sara Kiesler, Robert Kraut, Paul Resnick, and Aniket Kittur. 2010. Regulating behavior in online communities. *Evidence based social design: Mining the social sciences to build online communities* (2010), 125–179.

[36] Srijan Kumar, William L Hamilton, Jure Leskovec, and Dan Jurafsky. 2018. Community interaction and conflict on the web. In *Proceedings of the 2018 world wide web conference*. 933–943.

[37] Shamanth Kumar, Reza Zafarani, and Huan Liu. 2011. Understanding user migration patterns in social media. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.

[38] Charlotte Lambert, Ananya Rajagopal, and Eshwar Chandrasekharan. 2022. Conversational Resilience: Quantifying and Predicting Conversational Outcomes Following Adverse Events. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 16. 548–559.

[39] Cliff Lampe and Paul Resnick. 2004. Slash (dot) and burn: distributed moderation in a large online conversation space. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 543–550.

[40] Hanlin Li, Brent Hecht, and Stevie Chancellor. 2022. Measuring the Monetary Value of Online Volunteer Work. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 16. 596–606.

[41] J Nathan Matias. 2019. The civic labor of volunteer moderators online. *Social Media+ Society* 5, 2 (2019), 2056305119836778.

[42] Kevin Munger. 2017. Tweetment effects on the tweeted: Experimentally reducing racist harassment. *Political Behavior* 39, 3 (2017), 629–649.

[43] Edward Newell, David Jurgens, Haji Mohammad Saleem, Hardik Vala, Jad Sassine, Caitrin Armstrong, and Derek Ruths. 2016. User migration in online social networks: A case study on reddit during a period of community unrest. In *Tenth International AAAI Conference on Web and Social Media*.

[44] Bailey Poland. 2016. *Haters: Harassment, abuse, and violence online*. U of Nebraska Press.

[45] Ashwin Rajadesingan, Paul Resnick, and Ceren Budak. 2020. Quick, community-specific learning: How distinctive toxicity norms are maintained in political subreddits. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 14. 557–568.

[46] Sarah T Roberts. 2016. Commercial content moderation: Digital laborers' dirty work. (2016).

[47] Richard Rogers. 2020. Deplatforming: Following extreme Internet celebrities to Telegram and alternative social media. *European Journal of Communication* 35, 3 (2020), 213–229.

[48] Morgan Klaus Scheuerman, Stacy M Branham, and Foad Hamidi. 2018. Safe spaces and safe places: Unpacking technology-mediated experiences of safety and harm with transgender people. *Proceedings of the ACM on Human-computer Interaction* 2, CSCW (2018), 1–27.

[49] Morgan Klaus Scheuerman, Jialun Aaron Jiang, Casey Fiesler, and Jed R Brubaker. 2021. A framework of severity for harmful content online. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–33.

[50] Anna Schmidt and Michael Wiegand. 2019. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, April 3, 2017, Valencia, Spain*. Association for Computational Linguistics, 1–10.

[51] Joseph Seering, Geoff Kaufman, and Stevie Chancellor. 2022. Metaphors in moderation. *New Media & Society* 24, 3 (2022), 621–640.

[52] Joseph Seering, Robert Kraut, and Laura Dabbish. 2017. Shaping pro and anti-social behavior on twitch through moderation and example-setting. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*. 111–125.

[53] Joseph Seering, Tony Wang, Jina Yoon, and Geoff Kaufman. 2019. Moderator engagement and community development in the age of algorithms. *New Media & Society* 21, 7 (2019), 1417–1443.

[54] C Estelle Smith, Bowen Yu, Anjali Srivastava, Aaron Halfaker, Loren Terveen, and Haiyi Zhu. 2020. Keeping community in the loop: Understanding wikipedia stakeholder values for machine learning-based systems. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.

[55] Nathan TeBlunthuis, Benjamin Mako Hill, and Aaron Halfaker. 2021. Effects of algorithmic flagging on fairness: quasi-experimental evidence from Wikipedia. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (2021), 1–27.

[56] Jessica Vitak, Kalyani Chadha, Linda Steiner, and Zahra Ashktorab. 2017. Identifying women's experiences with and strategies for mitigating negative effects of online harassment. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. 1231–1245.

[57] Emily A Vogels. 2021. The state of online harassment. *Pew Research Center* 13 (2021).

[58] Donghee Yvette Wohn. 2019. Volunteer moderators in twitch micro communities: How they get involved, the roles they play, and the emotional labor they experience. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–13.

[59] Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th international conference on world wide web*. 1391–1399.

[60] Amy X Zhang, Grant Hugh, and Michael S Bernstein. 2020. PolicyKit: building governance in online communities. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 365–378.

[61] Justine Zhang, Jonathan P Chang, Cristian Danescu-Niculescu-Mizil, Lucas Dixon, Yiqing Hua, Nithum Thain, and Dario Taraborelli. 2018. Conversations gone awry: Detecting early signs of conversational failure. *arXiv preprint arXiv:1805.05345* (2018).

[62] Haoti Zhong, Hao Li, Anna Cinzia Squicciarini, Sarah Michele Rajtmajer, Christopher Griffin, David J Miller, and Cornelia Caragea. 2016. Content-Driven Detection of Cyberbullying on the Instagram Social Network.. In *IJCAI*, Vol. 16. 3952–3958.

A CONSTRUCTING THE SIMULATED SERVER FOR EVALUATION

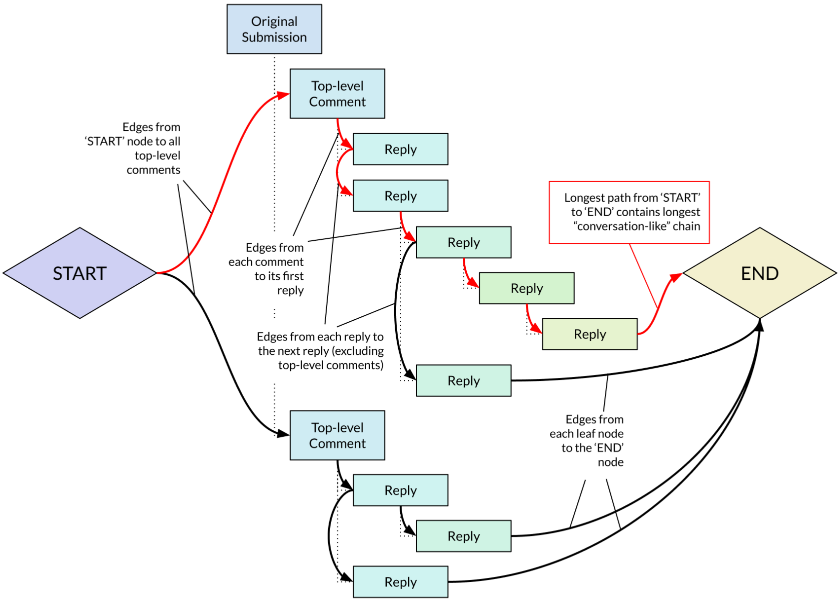

Original Submission
Top-level Comment
Edges from 'START' node to all top-level comments
Reply
Reply
START
Edges from each comment to its first reply
Reply
Longest path from 'START' to 'END' contains longest "conversation-like" chain
END
Reply
Reply
Edges from each reply to the next reply (excluding top-level comments)
Reply
Top-level Comment
Edges from each leaf node to the 'END' node
Reply
Reply
Reply

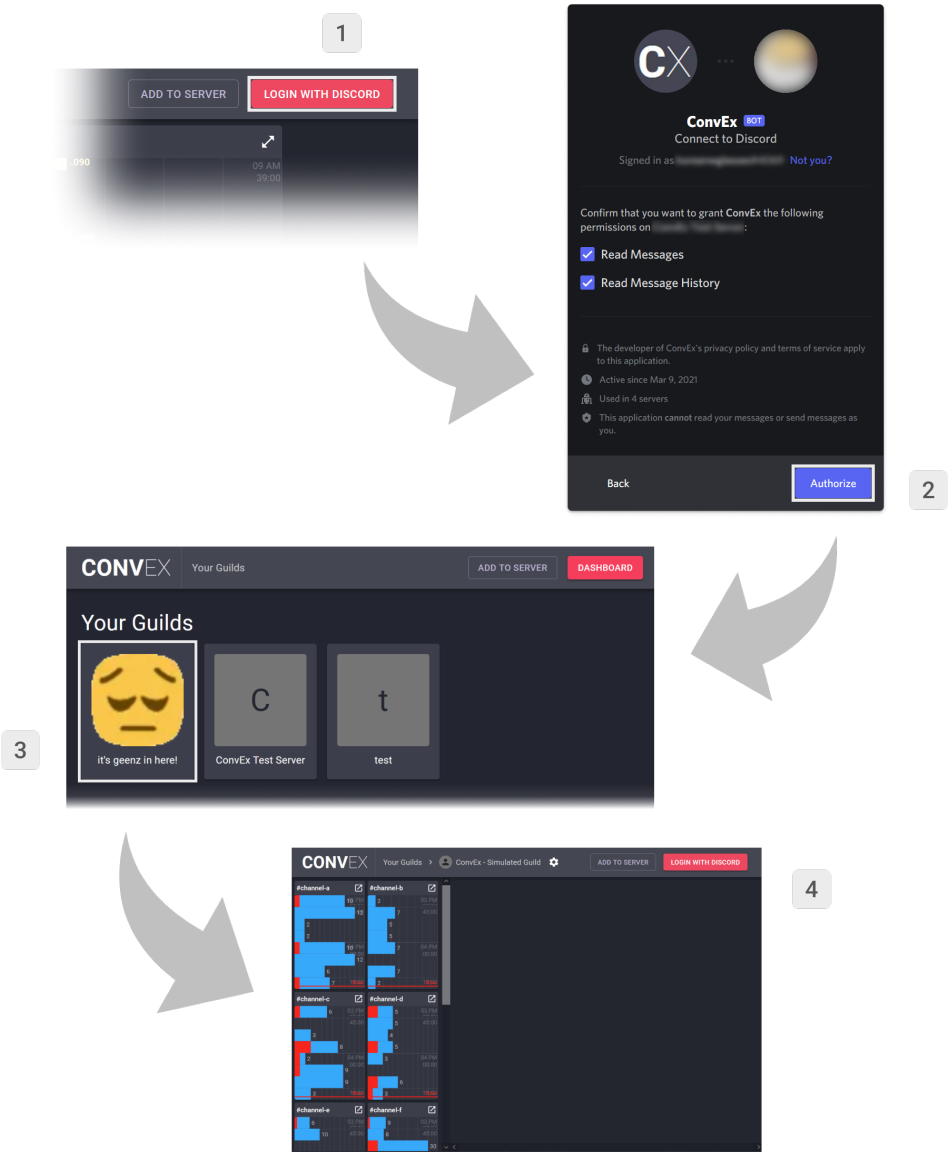
Fig. 9. An illustration of the directed-acyclic-graph constructed in the algorithm that was used to extract the longest "conversation-like" chain from top submissions on Reddit. Arrows indicate how the graph was constructed by creating edges between sibling comments and reply comments. Red arrows indicate the longest path along the graph from the START node to END node. This path represents the longest sequence of comments that are "conversation-like" in that adjacent comments are likely related in topic and have been written after reading the previous comment.

One of the challenges we ran into while recruiting was that potential participants were generally unwilling to add a bot from an unfamiliar third party to their Discord servers. Thus we decided to create a simulated server by sourcing conversations from public subreddits and "replaying" them directly onto ConvEx. Our method for extracting such conversations was to use the Reddit API to scrape the top submissions on the front page. For each submission, the full comment tree was fetched. From these trees, "conversation-like" chains of comments were extracted. Two comments are considered "conversation-like" if the later comment is either a direct reply to the previous comment (child) or a reply to the same comment as the previous comment (sibling). This captures the property of conversations where subsequent messages are likely related and were written after reading the previous message. This heuristic was used to capture a string of comments with a conversational nature.

To extract the longest such chain, the comment tree was represented as a directed acyclic graph, with edges connecting each comment to its replies, and edges connecting each comment to its next sibling. The “START” node had an edge to each top-level comment, and all nodes had an edge to the “END” node. A longest-path algorithm was used to find the longest path between “START” and “END”, and the comments along that path were extracted as the longest conversation-like chain (see Figure 9 for illustration). In this manner, 206 chains were extracted with a mean length of 20.77 comments and an average time of 1264 seconds (≈ 21 minutes) between comments.

To better simulate a real-time conversation setting the time interval between comments was reduced by a factor of 60, bringing the average time between comments to 21 seconds. In addition, the chains were separated into three groups based on the average frequency of messages sent to simulate channels with low, medium, and high levels of activity. In the simulated Discord server, there were 4 channels with low activity (> 19.5 seconds between messages on average), 3 channels with medium activity (10.1 - 19.5 seconds between messages on average), and 2 channels with high activity (1.86 - 10.1 seconds between messages on average). To simulate transition between topics, when a chain is exhausted, a longer delay (10x the average delay in the previous chain) is inserted before randomly selecting another chain with a similar activity rate.

B ADDITIONAL FIGURES



ConvEx BOT

Connect to Discord

Signed in as [redacted] Not you?

Confirm that you want to grant **ConvEx** the following permissions on [redacted]:

☑ Read Messages

☑ Read Message History

The developer of ConvEx's privacy policy and terms of service apply to this application.

Active since Mar 9, 2021

Used in 4 servers

This application **cannot** read your messages or send messages as you.

Back Authorize

Fig. 10. When a moderator first launches ConvEx, they are prompted to log in with their Discord account (1). This redirects them to Discord's authentication servers, obviating the need to attempt authentication ourselves, and improving security. The moderator is also prompted to authorize the ConvEx bot on their server, with a prompt clearly stating what data they are granting access to (2). Once the moderator is logged in, if they have added ConvEx to multiple guilds (i.e., servers), they can select which guild to view with ConvEx (3). Selecting a guild leads to the dashboard for that guild (4), whose features are summarized in Table 1.

Received July 2022; revised January 2023; accepted March 2023