压制赋权，容忍偏执：
审核 Twitch 平台上的仇恨言论管理

Prarabdh Shukla[*1], **Wei Yin Chong**[*2], **Yash Patel**[*1], **Brennan Schaffner**[2],
Danish Pruthi[1], **Arjun Bhagoji**[2, 3]
[1]Indian Institute of Science, [2]University of Chicago, [3]Indian Institute of Technology, Bombay
Correspondence: arjunp@iitb.ac.in

Abstract

为了满足内容审核的需求，在线平台开始依赖自动化系统。较新的实时互动形式（例如，用户对 Twitch 等平台的直播进行评论）对这些审核系统的延迟性提出了更高的要求。尽管这些系统相当普遍，但人们对其效果了解相对较少。在本文中，我们对 Twitch 的自动化审核工具（AutoMod ）进行了审查，以研究其在标记仇恨内容方面的有效性。为了进行审查，我们创建了一些流媒体账户作为独立的测试平台，并使用 Twitch 的 API 与实时聊天界面对接，以发送从 4 数据集中收集的 107,000 条评论。我们测量了 AutoMod 在标记明显的包含厌女症、种族主义、能力歧视和恐同的仇恨内容中的准确性。我们的实验表明，大部分仇恨信息，某些数据集高达 94% 的内容，能够绕过审核。将侮辱性词汇上下文性地添加到这些信息中导致 100% 的删除，揭示了 AutoMod 依赖侮辱性词汇作为审核信号的特性。我们还发现，与 Twitch 的社区指南相反，AutoMod 最多阻止了在教育或赋能环境中使用敏感词的 89.5% 无害示例。总体而言，我们的审查指出了 AutoMod 能力中的巨大差距，并强调了这些系统有效理解上下文的重要性。

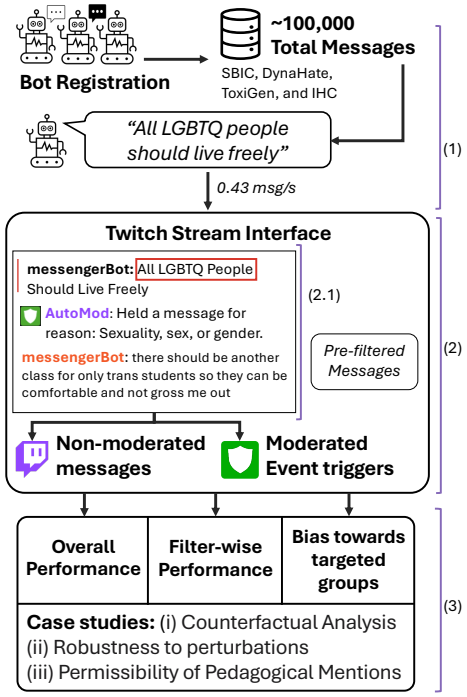


Figure 1: 审核流程：(1) 设置机器人与 AutoMod 交互并收集数据，(2) 大规模记录审核决策，(3) 分析 AutoMod 审核决策。(2.1) 显示了我们在 Twitch 界面实验中实际存在的审核和未审核消息实例。审核决策清晰地展示了 AutoMod 的局限性。

1 引言

为了使任何在线平台存在而不被充斥仇恨、色情、虐待、厌女和暴力内容，它必须对用户发布的内容进行审核 (Gillespie, 2018) 。除某些地区外 (Bundesamt für Justiz, 2022) ，很少或没有法律法规规定平台上可接受的内容 (Schaffner et al., 2024) 。此外，在全球北部，平台通常享有免除因托管用户生成内容而承担责任的法律保护 (47 U.S.C. ğ 230, 1996; EU, 2000) 。这种有利的监管框架赋予平台按其意愿审核内容的自由。关于这种自由的好处，讨论存在分歧，有些人赞扬其促进了在线平台的增长 (Kosseff, 2019) ，而另一些人则批评它助长了有害内容的传播 (Wakabayashi, 2019) 。平台通过服务条款和社区指南规范其用户的预期行为。虽然各个平台的政策和指南可能会有很大不同，但大多数都承诺为用户提供一个没有在线伤害的安全空间。例如，视频流媒体平台 Twitch 的社区指南中指出：

> *"Twitch does not permit behavior that is motivated by hatred, prejudice or intolerance, including behavior that promotes or encourages discrimination, denigration, harassment, or violence based on the following protected characteristics: race, ethnicity, color,... We also provide certain protections for age."*

[*] Equal contribution.

尽管有修辞上的避讳，施加在内容审核系统上的压力从未如此之大。由于发布内容的规模和速度以及对延迟的严格要求，各大平台在其审核流程中越来越多地开始集成自动化的，通常是基于机器学习的系统 (Gorwa et al., 2020) 。内容审核的实践还面临着平衡阻止广泛不当内容与维护用户言论自由的竞争目标的挑战。为应对这些压力，一些平台发布了有关有害内容状态及相应措施的汇总数据 (Twitch, 2024b; X, 2024; Meta, 2024) 。尽管有这样的透明度努力，人们对用于审核的底层算法及其可能引入的偏见知之甚少。

在这项工作中，我们对 Twitch 的内容监管系统进行了审查。Twitch 是一个数字平台，主要用于内容直播，内容在频道或“流”中创建，访问者可以通过基于文本的实时聊天观看和互动。我们关注仇恨言论内容，因为它在社会上非常重要，并且比其他有害内容（如错误信息）更容易定义。我们将 Twitch 识别为一个富有潜力的审查平台，因为它提供了三个关键优势：(i) Twitch 被广泛使用；(ii) Twitch 上的流媒体可以被“隔离”，使我们能够设置只有研究团队可以观看被测试内容的控制实验；(iii) 该平台为主播提供了一套称为 AutoMod 的基于机器学习的监管工具，允许主播控制不同类别的有害内容。每个类别都有选项可以阻止针对不同身份群体的有害内容，并可以调节监管的程度。通过这次审核，我们试图回答以下关键研究问题（§4 ）：(i) AutoMod 在标记充满仇恨的评论时的效果如何？(ii) 各个过滤器在阻止不同种类和意图的仇恨时有多具体和有效？以及 (iii) 不同目标群体的审核率是否一致，或者某些群体是否受到不成比例的影响？我们通过个案研究（§5 ）来细化这些问题，具体地说：(i) 审核过滤器对显式污辱的依赖程度如何；(ii) 过滤器如何应对关于弱势群体的敏感但具有教育性质的内容？以及 (iii) 知悉过滤器的恶意用户能在多大程度上影响它们的表现？为回答这些问题，我们开发了一个框架，允许我们在一个孤立的沙箱中大规模地对 AutoMod 进行压力测试，通过部署聊天机器人。在我们的研究中，我们使用了四个数据集——一个真实世界的评论数据集（SBIC），一个真实世界的隐性仇恨数据集（IHC），一个关于隐性仇恨的合成数据集（ToxiGen），以及一个设计来欺骗仇恨分类器的合成数据集（DynaHate）。我们在 **??** 中讨论了所使用的仇恨言论数据集，并在 **??** 中讨论了我们的实验设置。

我们的审核显示，Twitch 的自动化审核远未达到充分的程度（§4.1 ），即使在最严苛的设置下也仅标记出 22% 的仇恨内容。我们观察到，与种族、民族和宗教相关的仇恨样本最不容易被标记，仅有 12.3% 的仇恨例子被捕捉到。此外，在某些数据集中，我们发现针对心理残疾人士的仇恨内容有大量的 98% 逃过了审核。在我们使用的两个隐性仇恨数据集中，AutoMod 仅能标记出 6.8% 的仇恨例子，这意味着过滤器严重依赖于基于污言秽语的管理，且遗漏了隐性仇恨。令人担忧的是，关于社区的赋权或正面短语（例如，Figure 1 ）也被标记了出来，我们发现 AutoMod 对语义保持不变的扰动非常脆弱（见 §5 ）。

尽管不可否认，Twitch 为用户提供了强大且可定制化的管理工具，但我们的审核作为一个重要提醒，表明这些工具必须经过全面测试，并且其局限性应被明确指出。像我们这样的第三方审核可以通过提供关于全面管理挑战的定量证据，来丰富关于内容管理的讨论。我们希望我们的方法能激励并被普遍应用于审核自动化内容管理以及跨平台和数据模态的其他决策系统。

2 相关工作

内容审核 鉴于其重要性，内容审核已被广泛研究 (Keller et al., 2020) 。先前的工作系统地分析和批判了来自各种在线平台的审核政策，有的专注于单一平台 (Chandrasekharan et al., 2018; Fiesler et al., 2018; Keegan and Fiesler, 2017) ，有的则是涵盖众多平台的行业整体分析 (Schaffner et al., 2024) 。其他工作则集中于审核的具体方面，比如用户对审核的反应 (Cai et al., 2024; Ribeiro et al., 2023) ，审核对社区行为的影响 (Chancellor et al., 2016; Chandrasekharan et al., 2017; Chang and Danescu-Niculescu-Mizil, 2019) ，以及审核对视障用户的负面影响不成比例的问题 (Lyu et al., 2024) 。先前的工作还提出使用各种 AI 模型来辅助自动化内容审核 (Kumar et al., 2024; Franco et al., 2023a; Kolla et al., 2024a; Gray and Suzor, 2020) 。关于当前内容审核工作的深入讨论，我们建议读者参考 (Arora et al., 2023b) 。我们的工作通过审计现实世界中的自动化内容审核算法，补充了现有的研究。

审计算法 审计，如 (Gaddis, 2018) 所定义，是一种在实地环境中部署随机对照实验的方法。当审计针对算法和计算机系统时——称为“算法审计” (Sandvig et al., 2014; Metaxa et al., 2021b) ——则通过在输入中进行微小的更改来分析系统的输出，从而获得关于整个系统的见解。算法审计通常调查系统的潜在偏见和歧视行为 (Edelman and Luca, 2014; Speicher et al., 2018; Chen et al., 2018a; Metaxa et al., 2021a) 。其他研究则审计平台是否有效且公平地执行

其政策，例如针对 Facebook 和 Google 的政治广告政策 (Pochat et al., 2022; Matias et al., 2021)。算法审计涉及广泛的领域，审查了几个平台，包括住房（如 AirBnb 等租赁平台）(Edelman and Luca, 2014)、共享出行（Uber）(Chen et al., 2015)、医疗保健 (Obermeyer et al., 2019)、就业和招聘 (Chen et al., 2018b; Speicher et al., 2018)、广告 (Speicher et al., 2018) 和产品定价 (Mikians et al., 2012)。在像我们这样的典型算法审计中，审计人员仅能以黑盒方式访问系统，需要在这种访问级别下得出结论 (Cen and Alur, 2024)。一项同期发表的研究 (Hartmann et al., 2025) 评估了最近出现的各种审核 API（例如 OpenAI 的审核 API (Markov et al., 2023)）。该研究中所选择的数据集和方法与我们的相似，进一步验证了我们的方法。(Hartmann et al., 2025) 指出，大多数被评估的 API 对隐性仇恨内容的审核不足，而对教学/赋权内容、回收的辱骂词汇和对抗性言论的审核则过度。他们还强调审核 API 常常依赖于如“黑色”这样的群体身份关键词来做出审核决策。对于 AutoMod，我们几乎得出了相同的观察结果（见 §4.1，§5），这些结果共同指出了 SoTA 语言技术的能力与用于商业审核的技术之间的整体差距（见图表 2）。

在 Twitch 上直播 随着 Twitch 的普及，它成为了近期多项研究的主题，包括通过建模不同行为主体组的角色，将其视为新兴的政治空间 (Ruiz Bravo and Roshan, 2022)。另一项研究通过分析志愿版主的招聘、动机及其与其他在线平台的角色比较，研究了 Twitch 上的志愿版主 (Seering and Kairam, 2022)。最近的研究揭示了 2021 年在 Twitch 上经历的一系列攻击（被大众媒体称为“仇恨袭击”）是基于创作者的人口统计学而进行的有针对性的攻击 (Han et al., 2023)。据我们所知，我们是第一批研究 Twitch 自动化内容审核的研究者。

3 方法论

在本节中，我们首先描述我们对在线平台的调查。然后，我们描述用于审计的数据集，接下来是对审计的数学描述。

我们为选择审计平台设定了两个要求：（i）减少伤害：作为审计一部分发布的内容应仅对一个受控群体可见（即“隔离”）；（ii）高级审核工具：一套可配置的审核工具（最好利用原生机器学习模型）。第一个要求源于伦理要求，以最小化对毫无防备的用户造成的伤害，并且不助长现有的仇恨言论泛滥。机器学习系统往往有偏见，并且常常有可表征的缺陷，这使得审计使用该技术的平台变得很重要，因此有了第二个要求。此外，虽然一些平台可能采用以黑名单形式存在的较不先进的审核系统，但审计此类系统仅会测试名单的全面性，而不是提供的工具有多有效。我们调查了 43 个最大的用户生成内容平台，以查看是否有符合我们要求的，发现了两个合适的平台——Reddit 和 Twitch。我们也单独考虑了 Discord 作为候选平台，但最终选择 Twitch 进行此次审计，因为它的审核系统特别可配置，并且还揭示审核理由。我们根据我们的要求比较了这三个平台。

Twitch 提供了哪些审核工具？ 在 Twitch 上有三种主要的交互方式：主播与观众，观众与主播，观众与观众，这些交互中的内容都受到监管。在主播与观众的层面，监管是通过基于机器学习的工具进行的，这些工具检查视听流是否违反 Twitch 的平台政策。对视听内容监管的审计超出了本文的范围，但为未来的研究提供了一个有趣的方向（§6）。在这次审计中，我们重点关注通过实时聊天界面发生的观众与主播以及观众与观众之间的交互的监管。

主播以及指定的管理员可以配置 Twitch 的 AutoMod 工具，通过各种类型的过滤器自动处理可能大量涌入的聊天内容，以过滤掉不需要的内容。AutoMod 应用机器学习 (Twitch, 2024a) 来检测不需要的内容，并在管理员视图中显示这些内容（Figure 5）。此外，主播还可以创建一个词语或短语的黑名单，以限制聊天中出现这些内容。AutoMod 的可定制性使其可以根据检测力度和内容类别进行不同层次的调整，因此成为了一个有趣的审计测试平台。例如，侮辱性语言和歧视性内容的检测敏感度可以独立设定在 5 点的量表上（详情请参阅 **??** 关于我们如何为审计研究配置 AutoMod 的信息）。Twitch 还允许观众（独立于主播）开启聊天过滤器（Figure 7），以阻止某些内容类别，但这些过滤器并不像提供给管理员的选项那样详细。一个智能检测选项也在测试中，允许通过管理员的动作学习管理规则。由于这需要人工管理，是本文研究范围之外的内容。

在 Twitch 管理的问题内容类别中，大多数被管理的内容属于性行为、骚扰和仇恨行为类别（根据 Twitch 的 2024 透明报告 (Twitch, 2024b)）。我们将审计重点放在仇恨言论上，这更具体地归入 Twitch 管理的歧视 & 侮辱类别（表 4）。这一类别的言论在其意图和语言使用上通常是复杂的，使其成为一个有趣的研究对象。我们在实验中使用了四个数据集：DynaHate (Vidgen et al., 2021b)，社会偏见推理语料库（SBIC）(Sap et al., 2020)，ToxiGen (Hartvigsen et al., 2022) 和隐性仇恨语料库（IHC）(ElSherief et al., 2021)，其中 DynaHate 和 ToxiGen 是合

成生成的。在这些数据集中，前三个数据集附有指定目标群体的注释，使我们能够分层分析（§4.1）。对于 SBIC，每个示例都附有一个从 0 到 1 的攻击性评分，其中 0 代表最低攻击性评分，1 代表最高攻击性评分。我们使用 1 分的攻击性评分作为获取真实标签的阈值，除非有其他说明。更详细的数据集描述在 **??** 中提供。

符号和术语：我们将黑箱内容审核系统（如 AutoMod）定义为一个二元组 $\mathcal{S} = (\mathcal{F}, \mathcal{C})$。其中，$\mathcal{F} = \{\mathcal{F}_1, \ldots, \mathcal{F}_k\}$ 是一组 k 黑箱审核功能（过滤器），$\mathcal{C} = \{c_1, \ldots, c_k\}$ 是映射到现实世界概念（如残疾或厌女症）的对应抽象标准集，通常源自 $\mathcal{S}$ 试图执行的政策。每个 $\mathcal{F}_i : T \to \{0, 1\}$ 是一个将文本 $t \sim T$ 映射到标签 0（良性）或 1（违规）的函数，依据标准 c_i。这里 T 代表所有可能输入文本到审核系统的分布。输入 t 的审核决策由对应活跃过滤器集 $\mathcal{C}_A$ 的活跃审核功能 $\mathcal{F}_A$ 决定。$\mathcal{C}_A$ 是内容需要被审核的标准集。在 AutoMod 中，每个过滤器 $\mathcal{F}_i$ 进一步通过过滤级别 α 参数化——这是一个调节执法严格程度的离散度量。

在自动化内容审核中，集合 $\mathcal{C}$ 的元素构成了整个平台的审核策略，而 $\mathcal{C}_A$ 代表了当前主播或审核者所作出的决定。当我们说某个特定的过滤器集合 $\mathcal{C}_A$ 被“开启”时，我们指的是激活的审核函数 $\mathcal{F}_A$ 为 $\mathbb{1}(\bigcup_{c_i \in \mathcal{C}_A} \mathcal{F}_i = 1)$，当任何过滤器返回 1 时，该函数返回 1。在这项研究中，我们着重审查子集 $\tilde{\mathcal{C}} = \{\text{Disability, SSG, Misogyny, RER}\}$ [1] 的审核功能，该子集对应于广义类别的歧视和辱骂（**??**）。我们的分析在 §4.1 中，通过相对于我们基本数据集 $\mathcal{D}$ 的不同子集来改变 $\mathcal{C}_A \subseteq \tilde{\mathcal{C}}$（以及相应的 $\mathcal{F}_A$）展开。

在我们的实验中，我们构建了 $\mathcal{D} = \left(\bigcup_{c_i \in \tilde{\mathcal{C}}} D_{c_i}\right) \bigcup D_{\text{benign}}$，其中 D_{c_i} 中的每个文本样本至少有一个真实标签映射到 c_i。D_{benign} 是与 $\mathcal{C}$ 中任何类别不对应的文本样本，$\mathcal{C}$ 是 Twitch 允许过滤的一组标准。根据真实标签，我们使用标准指标如精度和召回率来衡量过滤器在检测仇恨言论上的有效性，包括整体数据集、不同类别以及每个类别中针对不同目标群体的效果。

基于过滤和政策指导的文本生成： 我们使用的基础数据集包含一般的言论样本，无论是否是仇恨言论，这些样本并非专门用于调查 AutoMod 的审核功能的鲁棒性以及 Twitch 声明的政策和 AutoMod 操作之间的一致性。在 §5 中，我们进一步测试 $\mathcal{F}$，通过定制样本来进行隐性仇恨检测、情境意识以及对语义保留输入的鲁棒性。

[1] SSG 代表性别、性取向和性别认同，RER 代表种族、民族和宗教。

4 实验 & 结果

在 **??** 中，我们描述了允许我们大规模记录 Twitch 审核决策的实验设置。然后我们在 §4.1 中讨论了 AutoMod 在不同环境下的表现。这里，我们简单介绍我们的实验流程（Figure 1）的设置和组件（详细信息在 Appendix B 中）。为简便起见，除非另有说明，每个过滤器的级别都设置为 $\alpha = 4$，即“最大过滤”。Twitch 提供了注册和开发聊天机器人的功能，这些机器人通常用于帮助主播提高流媒体的参与度，比如观众奖励或流媒体捐赠。为了以编程方式和大规模发送消息，我们创建了通过 Twitch 的开发者控制台注册的聊天机器人，使用在线临时电话号码、个人电话号码和别名电子邮件帐户的组合。运行实验流程的一个实例需要三个认证的机器人：一个信使机器人、一个接收者机器人和一个 Pubsub 机器人。

我们使用消息传递机器人在遵守 Twitch 的聊天速率限制 (Twitch, 2025) 的情况下向聊天流发送消息，以防止消息重复或遗漏。然后，我们使用接收器机器人模拟观看聊天流的用户——并记录所有没有被审核的消息。为了观察经过审核的消息，我们让第三个机器人订阅 Twitch 的 Pubsub 事件，这些事件提供导致审核的问题片段的信息，以及诸如 Ableism、Misogyny、Racism 和 Homophobia（Figure 6）等内部分类标签。虽然这些类别在 AutoMod 文档中没有详细说明，但我们从 API 使用中推断出 AutoMod 文档中的内容类别与内部类别之间的单射关系。通过这些标签，我们能够研究 AutoMod 陈述的审核原因，并在 **??** 中进行进一步的类别特定分析。总的来说，我们在 2024 年 12 月至 2025 年 1 月间发送和记录了约 300,000 条消息的审核（非）活动以进行实验。尽管遵守了速率限制，我们推测 Twitch 针对欺诈活动的安全措施 (Twitch, 2024b) 导致我们的聊天机器人被反复禁止，这迫使我们创建了 30 个不同的开发者账户。此外，我们观察到第三类消息，这些消息既没有被接收器检测到，也没有被 pubsub 机器人检测到，表明在 Twitch 上这些消息有一个未记录的第三目的地。我们怀疑这些消息——鉴于大多数都包含特定的仇恨言论——在 Twitch 的审核金字塔的服务级别被截获（见 Figure 4a），因此在 AutoMod（发生在频道级别）之前被过滤掉。从此，我们称这些消息为预先过滤，并在本节后面讨论其影响。

4.1 结果

在本节中，我们首先对 AutoMod 的整体有效性进行分析。然后我们讨论逐滤波器的性能和滤波器的特异性，接着是目标组特定的结果。

我们将仇恨内容按照 **??** 中的描述传递给 Twitch，对于每个示例，获得一个二进制标签 $Y \in \{0, 1\}$，指示示例是否被审核（$Y = 1$）或未被审核（$Y = 0$）。在这里，$\mathcal{C}_A = \tilde{\mathcal{C}}$，意味着所有在歧视和侮辱下的类别都被考虑在内。然后我们将这些与真实标签进行比较，并测量准确率、精确率、召回率或真阳性率（TPR），以及 F1 分数。正如 §1 中所述，内容审核系统旨在平衡 i) 阻止仇恨内容和 ii) 保障言论自由这两个相互竞争的目标。$F1_{\text{TPR, TNR}}$ 是 TPR 和真阴性率（TNR）之间的调和平均数。它衡量了两个目标的同时优化。召回率反映了目标 i) 的表现。AutoMod 的整体召回率仅为 22%，这与训练于类似数据的开源分类器相比相当低（Sap et al., 2020），也与最先进语言模型的零样本性能相比（见图 2）也相对较低。AutoMod 在即便是最明显的仇恨真实世界数据上也表现不佳，仅标记出 SBIC 标注者一致标记为仇恨的内容中的 19 %。在两个隐性仇恨数据集 ToxiGen 和 IHC 上，我们观察到远低于其他数据集的召回率（分别为 6 和 7%），这表明 AutoMod 在识别隐性仇恨方面表现不佳，缺乏对语境的理解（我们在 §5 中进一步验证了这一点）。我们还评估了 Twitch 的聊天过滤器（在 **??** 中描述），我们的发现（见 Appendix B.3）表明它在 $\alpha = 4$ 处与 AutoMod 行为完全相同。因此，我们在本文的其余部分将重点放在 AutoMod 上。

冒犯性阈值：对于 SBIC，我们在冒犯性得分上使用了不同的阈值来获取真实标签。更高的阈值反映了人们在将一个例子分类为仇恨言论时更严格的共识。当我们放宽冒犯性得分的阈值，以考虑不同意见和更多细微的仇恨实例时，召回率进一步下降（Figure 10）。在不同阈值下的进一步结果见于 Appendix B.6。

FN/FP 分析：我们推测高 FNR（1− 回召）主要是因为隐性仇恨。为了验证这一点，我们使用 LLM 来识别假阴性中的脏话（见 Appendix B.5），发现 89.8% 的假阴性中没有脏话（即是隐性仇恨）。我们还注意到 AutoMod 在负类上的表现很好，这从高 TNR 中可以看出。在对 DynaHate 进行类似分析时——DynaHate 有相对较高的 FPR——我们发现近 73% 的假阳性包含脏话。这项分析提示了 AutoMod 在依赖脏话作为仇恨信号上的严重依赖（进一步细节见 §5）。

会话级别的上下文意识：虽然关于误报和漏报的分析暗示了在消息级别上非常有限的上下文意识，我们进行了另一项实验以确定 AutoMod 是否在会话级别上（即跨不同的消息传递到聊天中）具备任何上下文意识。为此，我们从 SBIC 中选择了 100 个仇恨和 100 个良性例子。我们先单独传递了 100 个仇恨例子并观察到 35 个被屏蔽了。接着，我们将同样的 100 个仇恨例子与 100 个良性例子交错排列并观察到相同的 35 个消息再次被屏蔽，这表明 AutoMod 并不具备会话级别的上下文意识。此外，我们还通过不同的例子排列测试了我们的框架，以确保结果的可重复性。我们观察到改变输入消息的顺序并不改变任何例子的审核决定。

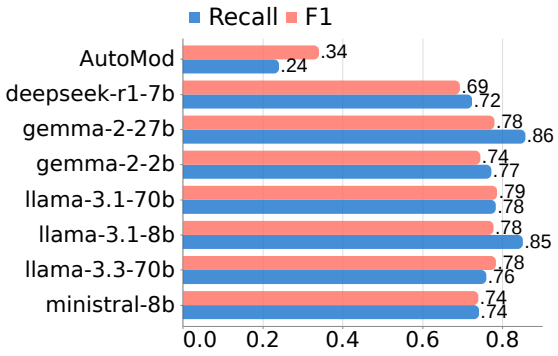


Figure 2: 将 AutoMod 与最先进（SoTA）的语言模型进行比较。语言模型被提示为包含 Twitch 社区指南的零样本指令。有关实验设置，请参见 Appendix B.8。

我们根据我们分析的四个过滤器手动对仇恨实例进行分类：残疾、厌女症、RER 和 SSG，从而构建 4 子集 $\mathcal{D}_{c_i} \subset \mathcal{D}$（见附录 D）。我们数据子集的质量控制分析在附录 B.10 中提供。此分析的主要目标是评估过滤器 $\mathcal{F}_i$ 在多大程度上能有效建模与其相关的标准 c_i。注意，我们可以获得任何被审核的消息的 AutoMod 内部类别，这允许我们确定即使在有多个过滤器启用的情况下是哪一个导致了审核。

过滤器召回率：首先，对于每个子集 $\mathcal{D}_{c_i}$，我们设置 $\mathcal{C}_A = \{c_i\}$ 来衡量过滤器的召回率（表 2）。我们观察到厌女症和性别、性和性别认同（SSG）过滤器具有最高的召回率。然而，考虑预过滤率也是很重要的。在所有数据集上，残疾和厌女症子集的预过滤率非常低。这意味着这些子集中的大多数例子都通过了过滤器，召回率很好地代表了仅过滤器性能。对于 SSG 过滤器，我们观察到相反的现象——整体预过滤率是 54.8 %，并且在各数据集上都很高。这意味着在检测与 SSG 相关的仇恨时，AutoMod 比其他类型的仇恨言论更重依赖于预过滤。然而，对于 RER 过滤器则不能得出类似的推论，

Dataset	Accuracy	Precision	Recall	TNR	$F1_{P,R}$	$F1_{TPR,TNR}$
SBIC (Sap et al., 2020)	0.73	0.42	0.19	0.91	0.26	0.31
DynaHate* (Vidgen et al., 2021a)	0.49	0.54	0.41	0.59	0.47	0.48
ToxiGen* (Hartvigsen et al., 2022)	0.53	0.86	0.07	0.98	0.13	0.13
IHC (ElSherief et al., 2021)	0.52	0.70	0.06	0.97	0.12	0.11
Overall	0.55	0.56	0.22	0.84	0.32	0.35

Table 1: AutoMod 的总体表现。AutoMod 在处理正常样本时相当准确，但在处理仇恨样本时则表现不佳。在 ToxiGen 和 IHC 数据集上的召回率低于其他数据集，这意味着 AutoMod 在检测隐含仇恨方面较弱。(*) 用于表示合成数据。绿色 表示指标的最佳值，红色 表示指标的最差值。

Dataset	SBIC		DynaHate		ToxiGen		Overall	
Filter	$\mathcal{R}$ (%)	Pf (%)	$\mathcal{R}$ (%)	Pf (%)	$\mathcal{R}$ (%)	Pf (%)	$\mathcal{R}$ (%)	Pf (%)
Disability	22.4	2.0	44.2	4.4	2.9	13.8	10.6	6.1
Misogyny	27.3	0.8	25.8	1.4	3.9	4.6	19.0	1.5
RER	17.3	36.1	20.5	18.8	6.6	21.5	12.3	22.0
SSG	32.0	64.1	25.3	55.3	5.7	44.3	17.5	54.8

Table 2: 不同数据集的过滤器召回率。我们报告了召回率 ($\mathcal{R}$) 和被预过滤的样本百分比 (Pf)。厌女症过滤器是最有效的。SSG 过滤器比其他过滤器更依赖于预过滤。

因为在 DynaHate 上 RER 过滤器的预过滤率几乎是 SBIC 的一半。不清楚数据本身缺乏多样性是否导致 SBIC 的预过滤率更高。我们推测 SBIC 有更多的种族歧视性语言出现在预过滤黑名单中，而 ToxiGen 则主要由隐性仇恨组成。

过滤器精确度：我们对每个 $\mathcal{D}_{c_i}$ 重复使用 $\mathcal{C}_A = \tilde{\mathcal{C}}$ 进行实验。这使我们能够测量过滤器的精确度 $P_{\mathcal{F}_i}$，即过滤器的特异性。一个简单的过滤器如果对于每一个仇恨文本都输出 1，会有很高的召回率，但这是不可取的，因为它筛选的例子超出了其标准——这在教育性或赋权性言论使用 n-gram 而这些 n-gram 常见于仇恨言论时可能不适用。因此，测量过滤器的特异性与召回率同样重要。过滤器的精确度如图 8 所示。我们观察到 RER 过滤器最精确，而厌女症过滤器最不精确。

在这项分析中，我们研究了 AutoMod 在阻止针对特定目标群体的仇恨行为方面的表现。为了获得针对不同目标群体的仇恨数据，我们使用数据集中目标群体的标签（映射细节见附录 D）。虽然每个子集可能对应于一个过滤器（例如，反黑人的种族主义 → RER，精神能力主义 → 残疾），但也可能存在一种情形，即一个例子可能与多个过滤器相关。为了解决在衡量目标群体表现时的问题，我们用 $\mathcal{C}_A = \tilde{\mathcal{C}}$ 评估召回率。通过开启所有过滤器，我们确保在这种情况下的召回率是对 AutoMod 能多好地阻止指向这些社区的仇恨的公平衡量。整个社区的召回率如图 3 所示。我们观察到，针对男性、黑人和精神残疾人士的仇恨言论比其他目标群体更有效地被 AutoMod 阻止。

预过滤和过滤级别 (α)： 我们分析了预过滤的示例，发现使用黑名单进行预过滤会导致目标群体之间的性能差异，这可能是由于黑名单构建中的偏差（见 Appendix B.9）。我们还在不同的过滤级别（α）评估 AutoMod，并发现对 $\alpha > 0$ 来说，性能（Appendix B.7）的提升（$+1.1\%$）是最小的。

5 案例研究：隐性仇恨、语境和稳健性

这一部分探索了三个案例研究，以检验 AutoMod 的多功能性和鲁棒性。

反事实分析： 与显性仇恨言论不同，隐性仇恨由于使用刻板印象、暗示或编码语言而更难被系统检测到——这些可能会避开传统的关键词过滤器。因此，检测此类内容需要对上下文、意图以及语言可以传播伤害的微妙方式有更深入的理解。对于本研究，我们从 SBIC 数据集中选择 110 个误报样本，并手动审查它们以确保没有任何侮辱性语言，同时仍然具有攻击性。然后我们通过编程将与人口统计群体相关的词语替换为与同一群体相关的侮辱性词语（例如，将“黑人”替换为 n 字）。当这些反事实示例被传递给 AutoMod （与 $\mathcal{C}_A = \tilde{\mathcal{C}}$），我们观察到 100 % 的召回率。我们的一些反事实集示例在表格 6 中展示。这项研究突出了 AutoMod 严重依赖使用侮辱性语言作为仇恨的标志以及

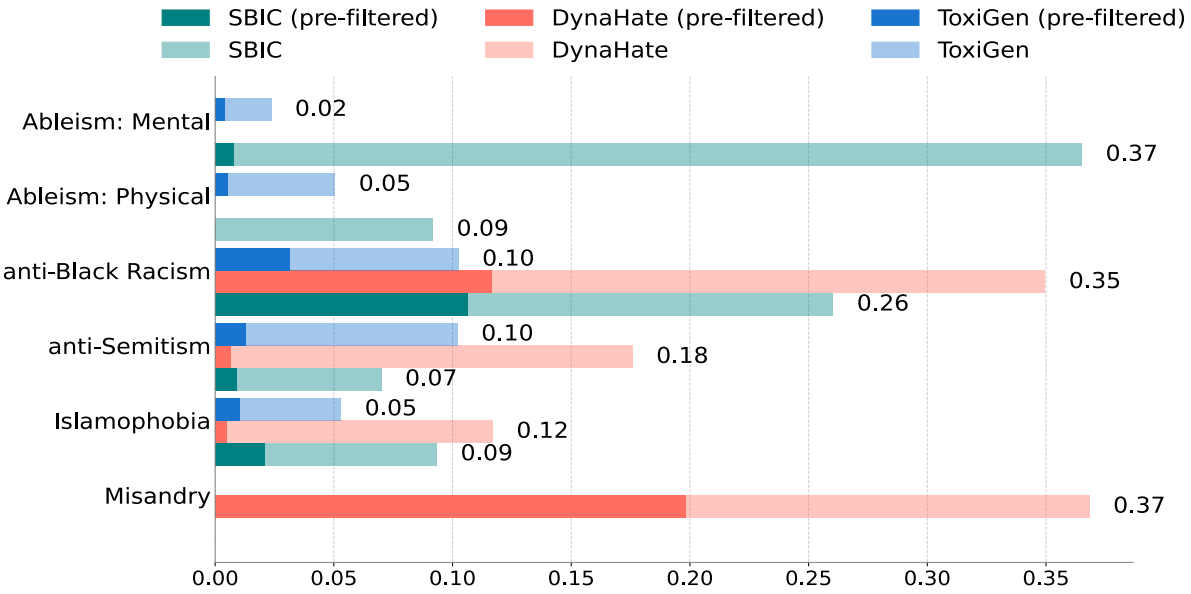


Figure 3: AutoMod 对不同社区的仇恨实例子集的召回率（所有的过滤器在歧视 & 侮辱类别下被启用）。每个柱状图的不透明部分表示被预过滤的信息的比例。

对上下文理解的不足，这使得隐性仇恨容易逃避检测。

Twitch 的社区准则明确承认了在内容审核中情境的重要性：

为研究 AutoMod 对这一政策的遵循情况，我们从 SBIC 中手动选择了 20 个非侮辱性敏感片段，并提示 GPT-4o 模型以不冒犯的方式生成包含这些片段的语句，包括在教育或赋权背景下使用这些术语。用于生成的提示和一些例子在 Appendix C.1 中提供。我们测试了这些例子（将 $\mathcal{C}_A$ 设置为 $\tilde{\mathcal{C}}$），并针对每个过滤器在某些过滤（$\alpha = 2$）和最大过滤（$\alpha = 4$）之间变化过滤级别。在前者情况下，AutoMod 标记了 89.5 % 的例子，而在后者设置中，它标记了 98.5 % 的例子。这种行为与上述政策形成对比。虽然主播可以手动配置 AutoMod 允许在流中可能激发这些片段在无冒犯语境中使用时，但这可能给恶意行为者提供机会继续散布仇恨。这项研究进一步强调了需要具备语境意识的自动化审核。

我们测量 AutoMod 在输入文本中进行具有语义保持的微小扰动时的鲁棒性，鉴于其似乎依赖于直接侮辱性语言。在这项研究中，我们提示 GPT-4o 对来自 SBIC 数据集的 50 个手动选择的敏感片段进行微妙的更改（例如添加空格或标点，改变拼写等）。我们采用了 6 种扰动方法（在表 7 中列出）。对于每个片段，我们还生成一个使用原片段的例子（未扰动）。生成所用的提示和扰动方法的定义在 Appendix C.2 中提供。我们观察到召回率在我们的某些扰动中从 100% 下降到 4% 。这表明恶意行为者可以轻易规避 AutoMod 。然而值得注意的是，AutoMod 能够检测到某些常用术语（如 n 字）的所有扰动（除了反向扰动）。这表明对于高度敏感的术语有一定程度的鲁棒性。

6 讨论和未来工作

相比于最先进的语言模型（Figure 2）甚至 GPT-2 (Sap et al., 2020)，AutoMod 在各数据集中的召回率和 F1 分数表现较差。我们对假阴性和假阳性（§4.1 和 Appendix B.5）的分析以及案例研究（§5）共同解释了 AutoMod 表现不佳是由于缺乏对仇恨言论的上下文理解。不管怎样，Twitch 在开发 AutoMod 方面的努力是值得称赞的，因为它提供的灵活性是大多数其他平台所缺乏的。我们的审核旨在突出 AutoMod 能力的当前不足，并且我们提出善意的建议，以便 Twitch 能够解决这些问题以提高用户安全性。未来工作的几个方向中，最直接的是评估 Twitch 的文本“智能检测”功能，并将审核扩展到视听内容。考虑到其他平台和语言，特别是在多语言仇恨言论数据可用性的情况下，(Arora et al., 2023a) 具有关键重要性。利用例如模型重构攻击 (Tramèr et al., 2016) 进行逆向工程和基于转移和查询的黑箱对抗样本生成 (Demontis et al., 2019; Bhagoji et al., 2018) 的技术，可能实现更细致的黑箱系统审核。总之，我们希望我们的研究能作为进一步审核在复杂社会技术环境中运行的决策系统的蓝图。

7

局限性

尽管我们试图在评估 Twitch 上的仇恨言论监管方面做到全面，但仍然存在一些关键的限制。首先，我们的研究仅考虑了一个平台，没有将 AutoMod 与任何其他已部署的文本监管进行比较。在本研究开始时，我们未找到其他合适的候选者。然而，诸如 Mistral 的监管工具 (Mistral AI Team, 2024) 之类的替代方案已经出现，这将使得在其他开源 LLM 进行监管的研究背景下进行有趣的后续研究成为可能。与本研究同时发表的一项研究 (Hartmann et al., 2025) 对类似数据集上各种监管 API 进行了类似评估。其次，我们在非对话环境中调查了充满仇恨和良性信息的消息，即，消息是依次传递的而没有任何对话模拟。需要更多工作来策划包含仇恨和良性内容的对话数据集。对于这类数据集的策划，研究人员可以考虑采用 DynaHate 风格 (Vidgen et al., 2021b) 的创建协议，这现在由于我们的框架而成为可能，该框架允许大规模查询像 AutoMod 这样的黑箱系统。此外，研究群体内/群体外动态可能揭示基于群体归属的内容差异待遇和监管。在 AutoMod 的情况下，我们推测我们的结果将保持不变，因为政策违规案例研究已经显示缺乏上下文意识。第三，我们的研究完全专注于英文文本。考虑其他语言和形式将是未来工作的重要方向。

在进行这项研究时，我们团队处理了大量攻击性内容。所有作者都意识到了工作的性质，并同意查看这些内容。需要注意的是，除作者外，没有其他人接触过这些材料，因为我们使用了隔离的实验流。此外，本文使用的有害内容来自开源数据集。

进行这项研究时，我们确实发布了违反 Twitch 服务条款的内容。然而，此行为是在 Sandvig v. Barr 案件中建立的法律界限内进行的 (Gilens, Naomi and Williams, Jamie, 2020) ，该案例保护此类研究活动。

潜在不利影响 我们展示了如何对攻击性文本的微扰绕过 Twitch 现有的审核系统，并可能产生负面影响，Twitch 用户可能利用这些例子。然而，这些技术和攻击已经有详细的文献记录，并在有关 NLP 分类器的文献中被广泛探讨。如果一个充满仇恨的行为者决心在 Twitch 上传播仇恨，他们不太可能不知道这些相对简单的微扰。

我们也承认不幸及不太可能的可能性，即我们的研究无意中增加了 Twitch 内部审核开发人员的负担。Twitch 已增加措施来保护其用户，我们不希望破坏这些努力。相反，我们打算在提倡对已部署系统进行仔细审核的同时，突出现有的不足。我们在本研究发布前已将我们的发现告知 Twitch，确保他们知悉其系统的漏洞，并可以采取适当措施。

8

致谢

我们感谢匿名审稿人提出的建设性建议。YP 对信实基金会研究生奖学金表示感谢，感谢其对其研究的支持。GC 感谢芝加哥大学计算机科学职业小型实习项目对她研究的支持。DP 另外感谢 Adobe 公司慷慨支持他团队的研究。我们指出，论文中提出的发现、结论和意见是作者的观点，并不一定反映赞助组织或机构的观点。

References

47 U.S.C. ğ 230. 1996. Protection for private blocking and screening of offensive material. Available at: https://www.law.cornell.edu/uscode/text/47/230.

Arnav Arora, Maha Jinadoss, Cheshta Arora, Denny George, Haseena Dawood Khan, Kirti Rawat, Seema Mathur, Shivani Yadav, Shehla Rashid Shora, Rie Raut, et al. 2023a. The uli dataset: An exercise in experience led annotation of ogbv. *arXiv preprint arXiv:2311.09086*.

Arnav Arora, Preslav Nakov, Momchil Hardalov, Sheikh Muhammad Sarwar, Vibha Nayak, Yoan Dinkov, Dimitrina Zlatkova, Kyle Dent, Ameya Bhatawdekar, Guillaume Bouchard, and Isabelle Augenstein. 2023b. Detecting harmful content on online platforms: What platforms need vs. where research efforts go. *Preprint*, arXiv:2103.00153.

Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. 2018. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *Proceedings of the European conference on computer vision (ECCV)*, pages 154–169.

Bundesamt für Justiz. 2022. Gesetz zur Verbesserung der Rechtsdurchsetzung in sozialen Netzwerken (Netzwerkdurchsetzungsgesetz –NetzDG).

Jie Cai, Aashka Patel, Azadeh Naderi, and Donghee Yvette Wohn. 2024. Content moderation justice and fairness on social media: Comparisons across different contexts and platforms. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI ' 24, page 1–9. ACM.

Sarah H. Cen and Rohan Alur. 2024. From transparency to accountability and back: A discussion of access and evidence in ai auditing. *Preprint*, arXiv:2410.04772.

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 CSCW*, CSCW '16, pages 1201–1213, New York, NY, USA. ACM.

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. *In Proceedings of ACM Human Computer Interaction*, 1(Computer-Supported Cooperative Work and Social Computing):31:1–31:22.

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):1–25.

Jonathan P. Chang and Cristian Danescu-Niculescu-Mizil. 2019. Trajectories of blocked community members: Redemption, recidivism and departure. In *Proceedings of WWW*, pages 184–195.

Le Chen, Ruijun Ma, Anikó Hannák, and Christo Wilson. 2018a. Investigating the impact of gender on rank in resume search engines. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–14, New York, NY, USA. Association for Computing Machinery.

Le Chen, Ruijun Ma, Anikó Hannák, and Christo Wilson. 2018b. Investigating the impact of gender on rank in resume search engines. In *Proceedings of the 2018 chi conference on human factors in computing systems*, pages 1–14.

Le Chen, Alan Mislove, and Christo Wilson. 2015. Peeking beneath the hood of uber. In *Proceedings of the 2015 internet measurement conference*, pages 495–508.

Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. 2019. Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In *28th USENIX security symposium (USENIX security 19)*, pages 321–338.

Benjamin Edelman and Michael Luca. 2014. Digital discrimination: The case of airbnb.com. *SSRN Electronic Journal*.

Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. 2021. Latent hatred: A benchmark for understanding implicit hate speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 345–363, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

EU. 2000. Articles 12-14, european union e-commerece directive.

Casey Fiesler, Jialun Jiang, Joshua McCann, Kyle Frye, and Jed Brubaker. 2018. Reddit rules! Characterizing an Ecosystem of Governance. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12.

Mirko Franco, Ombretta Gaggi, and Claudio E. Palazzi. 2023a. Analyzing the use of large language models for content moderation with chatgpt examples. In *Proceedings of the 3rd International Workshop on Open Challenges in Online Social Networks*, OASIS '23, page 1–8, New York, NY, USA. Association for Computing Machinery.

Mirko Franco, Ombretta Gaggi, and Claudio E. Palazzi. 2023b. Integrating Content Moderation Systems with Large Language Models. *ACM Trans. Web*.

S.M. Gaddis. 2018. *Audit Studies: Behind the Scenes with Theory, Method, and Nuance*. Methodos Series. Springer International Publishing.

Gilens, Naomi and Williams, Jamie. 2020. Federal Judge Rules It Is Not a Crime to Violate a Website ' s Terms of Service. [Online; accessed 19. Jan. 2025].

Tarleton Gillespie. 2018. *Custodians of the Internet: Platforms, Content Moderation, and the Hidden Decisions that Shape Social Media*. Yale University Press.

Robert Gorwa, Reuben Binns, and Christian Katzenbach. 2020. Algorithmic content moderation: Technical and political challenges in the automation of platform governance. *Big Data & Society*, 7(1):2053951719897945.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, and et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Joanne E Gray and Nicolas P Suzor. 2020. Playing with machines: Using machine learning to understand automated copyright enforcement at scale. *Big Data & Society*, 7(1):2053951720919963.

Catherine Han, Joseph Seering, Deepak Kumar, Jeffrey T. Hancock, and Zakir Durumeric. 2023. Hate raids on twitch: Echoes of the past, new modalities, and implications for platform governance. *Preprint*, arXiv:2301.03946.

David Hartmann, Amin Oueslati, Dimitri Staufer, Lena Pohlmann, Simon Munzert, and Hendrik Heuer. 2025. Lost in moderation: How commercial content moderation apis over- and under-moderate group-targeted hate speech and linguistic variations. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI ' 25, page 1–26. ACM.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326, Dublin, Ireland. Association for Computational Linguistics.

Jacob Foster. 2023. tmi.js. [Online; accessed 20. Jan. 2025].

Brian Keegan and Casey Fiesler. 2017. The evolution and consequences of peer producing wikipedia's rules. In *Proceedings of ICWSM*.

Daphne Keller, Paddy Leerssen, et al. 2020. Facts and where to find them: Empirical research on internet platforms and content moderation. *Social media and democracy: The state of the field and prospects for reform*, pages 220–251.

Mahi Kolla, Siddharth Salunkhe, Eshwar Chandrasekharan, and Koustuv Saha. 2024a. Llm-mod: Can large language models assist content moderation? In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI EA '24, New York, NY, USA. Association for Computing Machinery.

Mahi Kolla, Siddharth Salunkhe, Eshwar Chandrasekharan, and Koustuv Saha. 2024b. LLM-Mod: Can Large Language Models Assist Content Moderation? *CHI EA '24: Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–8.

Jeff Kosseff. 2019. *The twenty-six words that created the Internet*. Cornell University Press.

Deepak Kumar, Yousef AbuHashem, and Zakir Durumeric. 2024. Watch your language: Investigating content moderation with large language models. *Preprint*, arXiv:2309.14517.

Yao Lyu, Jie Cai, Anisa Callis, Kelley Cotter, and John M. Carroll. 2024. "i got flagged for supposed bullying, even though it was in response to someone harassing me about my disability.": A study of blind tiktokers' content moderation experiences. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI ' 24, page 1–15. ACM.

Todor Markov, Chong Zhang, Sandhini Agarwal, Tyna Eloundou, Teddy Lee, Steven Adler, Angela Jiang, and Lilian Weng. 2023. A holistic approach to undesired content detection in the real world. *Preprint*, arXiv:2208.03274.

J. Nathan Matias, Austin Hounsel, and Nick Feamster. 2021. Software-supported audits of decision-making systems: Testing google and facebook's political advertising policies. *Preprint*, arXiv:2103.00064.

Meta. 2024. Community Standards Enforcement Report. [Online; accessed 19. Jan. 2025].

Danaë Metaxa, Michelle A Gan, Su Goh, Jeff Hancock, and James A Landay. 2021a. An image of society: Gender and racial representation and impact in image search results for occupations. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–23.

Danaë Metaxa, Joon Sung Park, Ronald E. Robertson, Karrie Karahalios, Christo Wilson, Jeff Hancock, and Christian Sandvig. 2021b. Auditing algorithms: Understanding algorithmic systems from the outside in. *Foundations and Trendső in Human–Computer Interaction*, 14(4):272–344.

Jakub Mikians, László Gyarmati, Vijay Erramilli, and Nikolaos Laoutaris. 2012. Detecting price and search discrimination on the internet. In *Proceedings of the 11th ACM workshop on hot topics in networks*, pages 79–84.

Mistral AI Team. 2024. Mistral Moderation API. [Online; accessed 16 Feb. 2025].

Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. 2019. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453.

Victor Le Pochat, Laura Edelson, Tom Van Goethem, Wouter Joosen, Damon McCoy, and Tobias Lauinger. 2022. An audit of facebook's political ad policy enforcement. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 607–624, Boston, MA. USENIX Association.

Manoel Horta Ribeiro, Justin Cheng, and Robert West. 2023. Automated content moderation increases adherence to community guidelines. *Preprint*, arXiv:2210.10454.

Nadia Ruiz Bravo and Maryam Roshan. 2022. The political turn of twitch –understanding live chat as an emergent political space.

Christian Sandvig, Kevin Hamilton, Karrie Karahalios, and Cédric Langbort. 2014. Auditing algorithms : Research methods for detecting discrimination on internet platforms.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 2020. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5477–5490, Online. Association for Computational Linguistics.

Brennan Schaffner, Arjun Nitin Bhagoji, Siyuan Cheng, Jacqueline Mei, Jay L Shen, Grace Wang, Marshini Chetty, Nick Feamster, Genevieve Lakier, and Chenhao Tan. 2024. "community guidelines make this the best party on the internet" : An in-depth study of online platforms' content moderation policies. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI ' 24, page 1–16. ACM.

Joseph Seering and Sanjay R. Kairam. 2022. Who moderates on twitch and what do they do? quantifying practices in community moderation on twitch. *Proc. ACM Hum.-Comput. Interact.*, 7(GROUP).

Till Speicher, Muhammad Ali, Giridhari Venkatadri, Filipe Nunes Ribeiro, George Arvanitakis, Fabrício Benevenuto, Krishna P. Gummadi, Patrick Loiseau, and Alan Mislove. 2018. Potential for discrimination in online targeted advertising. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 5–19. PMLR.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pages 601–618.

Twitch. 2024a. How to Use AutoMod. [Online; accessed 19. Jan. 2025].

Twitch. 2024b. Transparency Reports. [Online; accessed 18. Jan. 2025].

Twitch. 2025. Chat & Chatbots. [Online; accessed 19. Jan. 2025].

Bertie Vidgen, Kayla Chatelon, Scott Hale, Helen Margetts, and Dong Nguyen. 2021a. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4216–4231. Association for Computational Linguistics.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2021b. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1667–1682, Online. Association for Computational Linguistics.

D Wakabayashi. 2019. Legal shield for websites rattles under onslaught of hate speech. *The New York Times, Aug*, 6.

X. 2024. Dsa transparency report - october 2024.

本附录提供了与主论文每一部分对应的附加信息，如下所示：

1. 平台选择和 Twitch 管理详情（Appendix A）：提供了 **??** 中提到的 Twitch 管理选项的更多细节。
2. 数据集详情（**??**）：在 **??** 中介绍的 4 个仇恨言论内容数据集的详细描述。
3. 实验设置详情（Appendix B）：详细介绍从 Twitch 发送和接收消息的步骤，扩展了 **??**。
4. 进一步结果（**??**）：基于 §4.1 中的结果，并分析来自 AutoMod 的假阴性和假阳性、SBIC 数据阈值、不同 AutoMod 过滤水平、预过滤偏差以及过滤器特定数据集的质量控制分析。
5. 消融详情（Appendix C）：关于在 §5 中用于构建反事实、策略遵循样本和扰动示例的方法的信息。
6. 滤波器和社区特定的子集提取（Appendix D）：§4.1 实验中用于获取子集的程序的数据集分解。

A 平台选择与 Twitch 审核细节

作为平台探索和 Twitch 管理部分（**??**）的扩展，我们提供了更多关于指导我们平台选择的调查、Twitch 界面及其内部文档的详细信息。

在 Table 3 中，我们提供了关于我们选择的 3 个平台在审计适用性方面的详细信息。尽管 Reddit 和 Discord 都提供了适合实验的孤立环境，但两者都没有提供用于攻击性文本检测的本机机器学习模型。

A.1 Twitch 自动化管理

在 Figure 4a 中，我们展示了 Twitch 管理金字塔 (Twitch, 2025)。如 **??** 中所述，我们专注于通过 AutoMod 和聊天过滤器管理的观众到主播和主播到主播的互动。在 Figure 4b 中，我们展示了 Twitch 为版主提供的各种级别的过滤强度定制选项。

Table 4 描述了 Twitch 的官方内容类别及其在 04 年 7 月 2024 的差异定义。审核的调查重点是 AutoMod 在歧视和辱骂内容类别方面的整体表现的有效性。更具体地说，所使用的数据集提供了与歧视和辱骂的各个子部分的真实情况有关的信息，如 Table 4 中的残疾、SSG、厌女症和 RER。

这个控制面板是可定制的，使得主播能够实时查看其流的 AutoMod 队列，并手动批准或拒绝如 Figure 5 中所示的检测到的信息。AutoMod 队列还显示关联的内容类别，以便告知主播 AutoMod 在审核背后的原因。高亮显示的部分还帮助用户专注于被审核信息中的问题片段。

在评估 AutoMod 在协调各种类型的仇恨内容方面的整体表现时，单纯的二元评估被认为是主要的，但不足以有效确定精确度和准确性方面的表现。因此，AutoMod 的审核原因相对于 Twitch 预定义的审核类别提供了更多的见解。为了以编程方式访问主播的 AutoMod 队列，使用了 Twitch Pubsub 事件订阅来抓取 API 结果，每当 AutoMod 在队列中记录一条消息进行审核时。然而，我们发现 AutoMod 的内部类别与 Twitch 平台上关于内容审核的文档和政策不匹配。Figure 6 展示了一幅来自 automodqueue 事件订阅的 Twitch API 结果快照，详细说明了消息及其元数据。该元数据包括被审核消息的 topics 和 category，我们依据此对类别进行语义映射。

所有使用的数据集都是开源的，并被允许用于研究。我们所有的数据都是英文的。

A.2 Dynahate

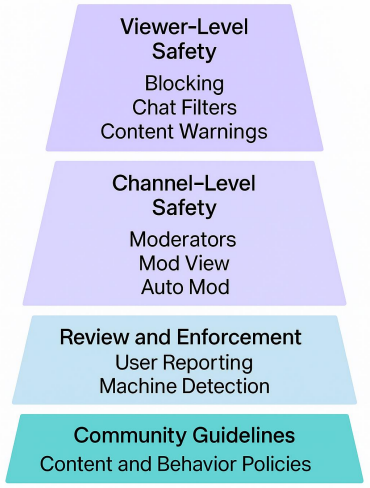
Dynahate 数据集 (Vidgen et al., 2021b) 包含大约 41,000 条目，每个条目被标记为 0（非仇恨）或 1（仇恨），并由受过训练的标注人员通过四轮动态数据创建生成和标记。该数据集包括 54 % 的仇恨示例。在我们的工作中，我们使用了 Dynahate 数据集中的三列：文本、标签和目标。‘目标’列指定了仇恨所指向的社区，使其特别适用于将数据分为过滤和特定社区的子集。

A.3 SBIC

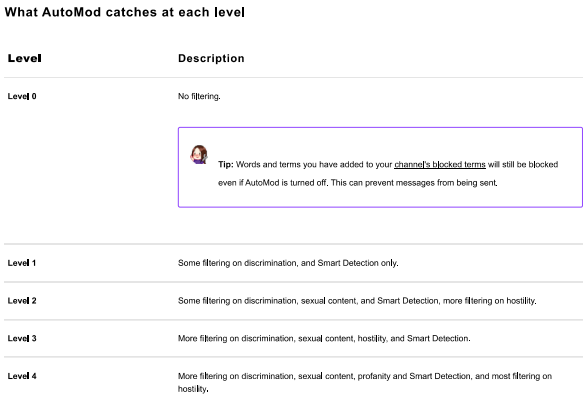
社会偏见推理语料库（SBIC）(Sap et al., 2020) 是一个大规模数据集，包含了 34,000 条社交媒体帖子中超过 15 万条注释，用于捕捉在线语言中的细微偏见和刻板印象。其关注于攻击性内容、隐含的刻板印象以及目标人群，这使得它尤其适合用于审计 Twitch 的审核系统，特别是在与群体性偏见和语境攻击性相关的领域。我们从 SBIC 训练数据中抽样了 2 万个例子用于我们的研究。由于 SBIC 提供了平均注释者攻击性评分，我们设定了一个阈值为 1（所有注释者对某个例子的攻击性达成了一致），来评估 AutoMod 的整体效果，结果在仇恨与非仇恨的比例为 1:3。为了进一步针对特定过滤器将其划分为仇恨子集，我们使用了 0.5 的阈值，总共得到了 8,748 个例子。SBIC 数据集中包含了许多特征，但在我们的分析中，我们使用了

Platform	Closed environment	Moderation tools
Discord	Private server with permissions for everyone but admin disabled	Image hashing and “ML powered tech" for child sexual abuse material; No native ML models for text, relies on keyword detection
Reddit	Via private subreddits	Needs plugins for ML models and subreddit specific rules for moderation; Well-investigated (Kolla et al., 2024b; Kumar et al., 2024; Franco et al., 2023b)
Twitch	Untagged streams are private	Customisable moderation levels using native ML models; Also has keyword lists and Smart Detection to train model on the actions of human moderators

Table 3: 候选平台的对比。我们调查了各种平台，并主要因为其可配置性和基于 ML 的审核工具选择了 Twitch。



(a) Twitch 的审核金字塔



(b) AutoMod 调节水平 (α) 从 0 到 4 级的区别

Figure 4: Twitch 官方的内容审核工具文档：（a）Twitch 对内容进行审核的不同层级及其各自的方法，（b）Twitch 关于 AutoMod 审核层级的文档（在这项工作中被称为 α ），范围从 0 到 4。

以下特征：帖子、目标少数群体、目标类别、offensiveYN（指示例子是否具有攻击性、非攻击性或模糊性），以及用于聚合攻击性得分的注释者相关列。

A.4 ToxiGen

ToxiGen 数据集 (Hartvigsen et al., 2022) 是一个大规模的数据集，包含约 27.4 万条关于 13 个少数群体的有毒和良性陈述。这个数据集对我们的工作特别有用，因为它包含了合成生成的隐性仇恨言论，可以让我们评估过滤器对隐性仇恨内容的有效性。作者使用 GPT-3 模型通过提供人工筛选的有毒和非有毒示例作为提示来生成隐性仇恨示例。有关数据生成方法的详细信息，我们建议读者参考 ToxiGen 论文。ToxiGen 包含了各种特性，但我们在工作中使用了以下特性：prompt，generation（模型生成的文本——我们用于审核的示例），prompt label，以及生成句子的 roberta 预测。对于我们的实验，我们从数据集中随机抽取了 2 万条仇恨示例，选择具有有毒提示且 roberta_prediction 分数在 0.8 到 1.0 之间的条目。同样，我们也抽取了 2 万条非仇恨示例，这些示例具有非有毒提示且 roberta_prediction 分数在 0 到 0.2 之间。

A.5 隐性仇恨语料库 (IHC)

隐性仇恨语料库（IHC） (ElSherief et al., 2021) 是一个包含大约 6,000 个隐性仇恨例子的数据库。在我们的实验中，我们额外从该数据库中随机选取了 6,000 个良性例子。它包括从仇恨

Moderation Category	Explanation of Hate Speech Category
Sexual Content	Words or phrases referring to sexual acts and/or anatomy.
Discrimination and Slurs	Includes race, religion, gender-based discrimination. Hate speech falls under this category.
I) Disability	Demonstrating hatred or prejudice based on perceived or actual mental or physical abilities.
II) SSG	Demonstrating hatred or prejudice based on sexual identity, sexual orientation, gender identity, or gender expression.
III)Misogyny	Demonstrating hatred or prejudice against women, including sexual objectification.
IV) RER	Demonstrating hatred or prejudice based on race, ethnicity, or religion.
Hostility	Provocation and bullying, sexual harassment.
Profanity	Expletives, curse words, and vulgarity. This filter especially helps those who wish to keep their community family-friendly.
Smart Detection	Detects unwanted messages (including spam) based on moderation actions taken in your channel.

Table 4: AutoMod 内容类别。AutoMod 管理的各种类别和子类别的定义。

社区及其 Twitter 追随者那里收集的仇恨言论的目标和隐含意义。在我们的工作中，我们使用这个数据集来评估所有歧视过滤器的有效性。由于数据集不够大，无法分割成特定过滤器的子集，我们仅仅移除那些与歧视不一致的例子。从这个数据集中，我们利用了帖子和目标列。

B 实验设置细节

在这里，我们详细描述构建实验流程 (Figure 1) 所需的设置和组件，该流程在 **??** 中已简要描述。实施的关键步骤包括：1) 机器人创建以发送消息，包括遵循安全措施、注册和范围处理，2) AutoMod 事件订阅用于将消息分类为已审核和未审核，以及 3) 结果处理。

B.1 以编程方式发送消息

发送大规模程序化消息需要以下步骤：

步骤 1：创建一个聊天机器人为了处理聊天机器人的创建，Twitch 要求每个账户都有经过双因素认证的电话号码和电子邮件地址。为了实现，创建的机器人使用了免费的在线电话号码进行 2FA、临时电话号码（仅限验证过的电话号码）以及个人电话号码。不同的机器人还需要在关联的已验证帐户的 Twitch 开发者门户中进行应用注册。此过程需要指定一个唯一的聊天应用名称、OAuth 重定向 URL 规范：https://localhost:3000，以及机器人的功能：Chat Bot 。该应用允许 Twitch 开发者提取特定于应用的 Client-ID 和 Client-Secrets；

步骤 2：认证和注册聊天机器人 Twitch 使用 Twitch OIDC 授权代码授权流程在 OAuth 2.0 下启用认证，以便授予应用程序对 Twitch HTTPS 资源的特定访问权限。此访问权限，例如，指定为 IRC 聊天机器人的 `chat:read` 和 `chat:write` 访问权限。当从 HTTPs POST 请求到 `https://id.twitch.tv/oauth2/token` 收到授权代码后，应用程序访问令牌和刷新令牌被检索并用于 API 调用。由于每个应用程序的访问令牌用于连接 Twitch 聊天和 AutoMod 队列，我们使用应用程序的刷新令牌自动处理这些授权令牌的更新，并在到达其过期时间时进行重新连接。

步骤 3：以适当的时间配置大规模发送消息我们通过包含一个信使机器人、频道机器人和 Pubsub 机器人（三机器人配置）创建一个实验管道实例。信使机器人发送最多 5 条消息，每条消息之间等待 4 秒。在每次迭代中，我们由于 Twitch 的聊天速率限制 (Twitch, 2025) 将暂停 3.5 秒，使普通聊天机器人账户每 30 秒发送少于 20 条消息。这确保了每条消息都被发送并处理到流聊天中，并不会因为连接延迟而导致重复或遗漏。我们进行了迭代测试，得出结论，这种暂停和消息数量的配置由于我们四个数据集中的大量消息允许较长时间的实验运行。

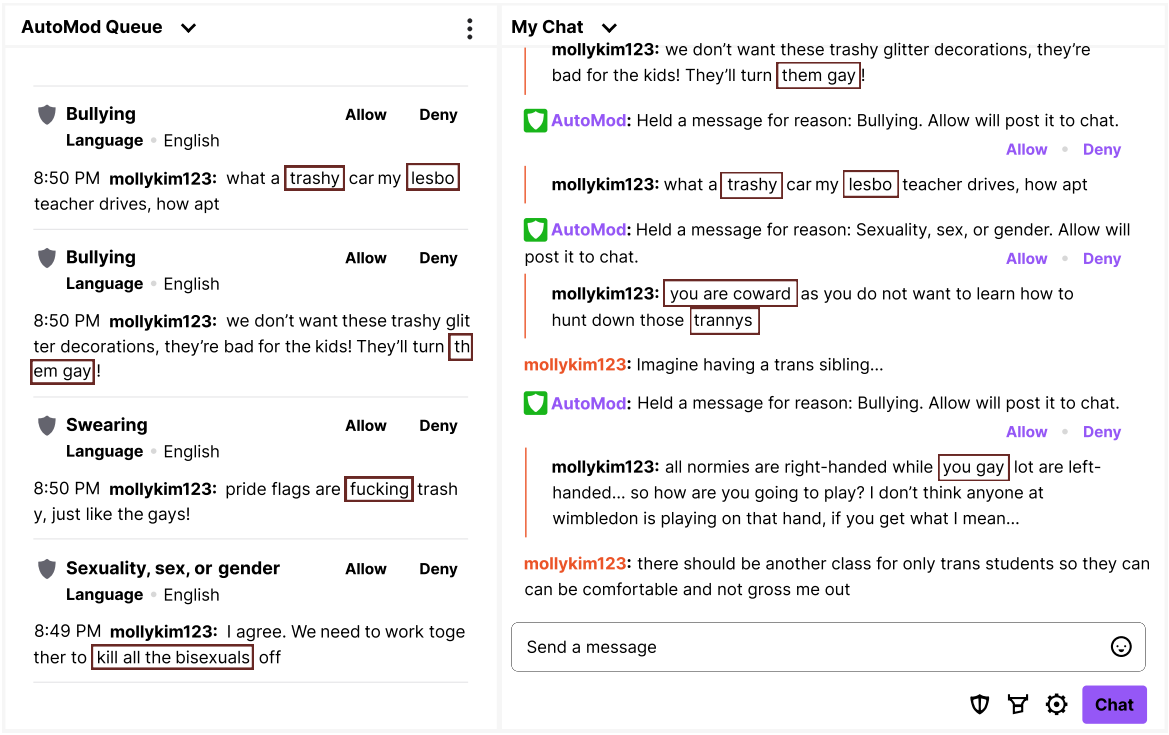


Figure 5: 从主播帐户看到的管理视图。（注意：原始截图是暗模式界面，为了打印清晰度手动调整为亮模式；与实际亮模式可能存在细微的视觉差异。）

B.2 接收审核和未审核的消息

为了接收和排序消息，我们需要创建额外的机器人并解析它们的输出。这种连接是通过使用 `tmi.js` JavaScript 包来实现的，该包使用 tmi.js 服务器创建与 Twitch IRC 服务器的连接 (Jacob Foster, 2023) 。Twitch IRC 提供了一个功能简化的 RFC1459 和 IRCv3 消息标签规范 (Twitch, 2025) ，用于解析到指定 Twitch 频道的消息。这种非管理消息的提取由接收器机器人处理。Pubsub 机器人使用 PubSub 事件订阅从 `AutoMod` 队列中实时提取消息，这些消息可以从频道创建者的仪表板访问。使用 twitchAPI.js，我们建立与 Twitch 机器提供 Pubsub 服务的 websocket 连接，注册的聊天机器人从中监听 `AutoMod` 队列项目，并在 websocket 框架中接收消息元数据。

从 Pubsub 接收到的 JSON 提供了有关被审核片段的信息，以及像 `Ableism`, `Misogyny`, `Racism`, and `Homophobia` (Figure 6) 这样的被审核内容的内部标签。虽然 `AutoMod` 文档中没有详细描述这些类别，但我们推断 `AutoMod` 文档中的内容类别与 API 调用中的内部类别之间存在一种单射关系。通过这些标签，我们可以调查 `AutoMod` 声明的审核原因，并在 **??** 中进行进一步的特定类别分析。

尽管没有超过速率限制，我们怀疑根据 2024 年透明度报告中所述，Twitch 上欺诈活动的增加导致了我们聊天机器人被不准确但经常性地永久封禁。因此，这导致创建了至少 30 个不同的账户。在进行若干初步实验并分析结果后，我们发现某些消息从未出现在接收机器人的收件箱中，因此也未触发任何发布/订阅事件。为了进一步调查，我们手动传递了一些这些消息，发现 Twitch 已经预筛选了它们，阻止了它们被发送。为了解决这个问题，我们增强了我们的代码，将所有消息——包括由 AutoMod 调节的和没有调节的——记录到 CSV 文件中。通过将这个日志与原始输入数据进行比较，并过滤掉缺失的消息，我们成功识别出了一组预筛选的消息。

我们所有关于语言模型的实验都是在一个配备 6 A6000 GPU 的工作站上进行的。

B.3 评估聊天过滤器

观众控制的聊天过滤器只能在广泛的歧视和诋毁类别的粒度上进行操作，并且不会触发用于检测管理事件的 Pubsub 机器人事件，这使得大规模实验变得困难，因为需要人工检查来识别管理事件，该事件由 “***” （星型模式）表示。为了审计 `AutoMod` 和聊天过滤器之间的一致性，我们随机抽样 200 个来自 SBIC 的例子，包含相同数量的已管理和未管理的例子，并通过聊天过滤器处理它们。我们观察到聊天过滤器无法管理任何未管理的例子（Figure 7

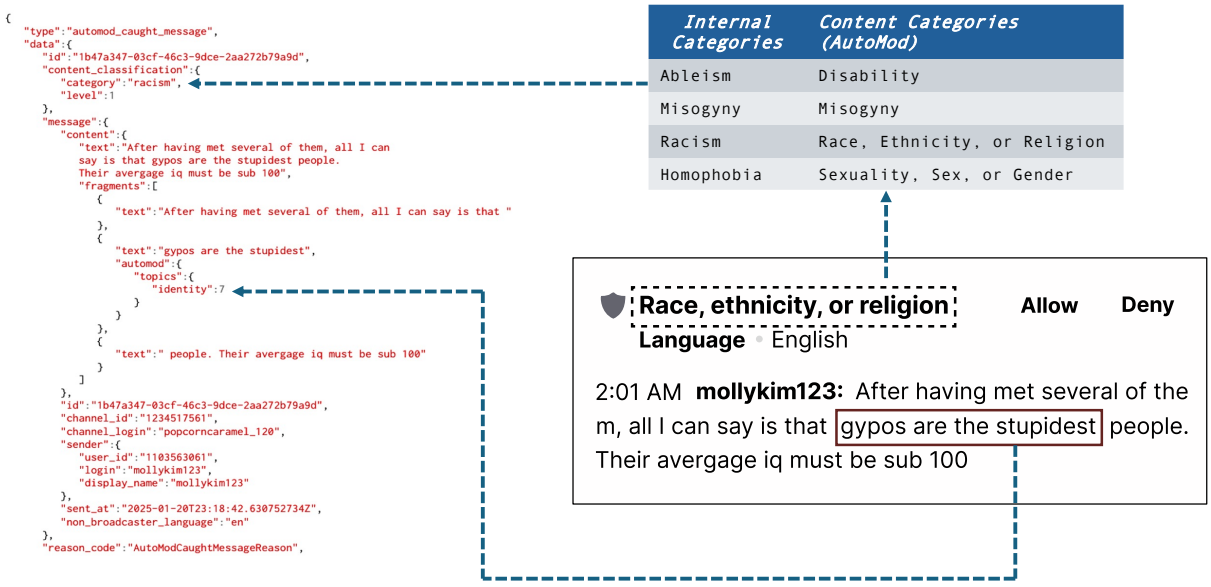


Figure 6: 将 PubSub 输出与审核决策关联：（左）从 PubSub 接收到的 JSON 输出示例；（右上）将 JSON 输出映射到 AutoMod 的内容类别（在 Table 4 中描述）；（右下）将 automodqueue （通过 UI 对主播可见）映射到从 PubSub 获得的 JSON。

）。我们推断，聊天过滤器在 $\alpha = 4$ 上的行为与 AutoMod 类似。

B.4 滤波精度

滤波器 $\mathcal{F}_i$ 的精度，以 $P_{\mathcal{F}_i}$ 表示，定义如下：

$$P_{\mathcal{F}_i} = \frac{\sum_{x \sim \mathcal{D}_{c_i}} \mathbb{1}(\mathcal{F}_i(x) = 1)}{\sum_{j \in \widetilde{\mathcal{S}_c}} \left(\sum_{x \sim \mathcal{D}_{c_j}} \mathbb{1}(\mathcal{F}_i(x) = 1) \right)}$$

它是指符合标准 c_i (即，从 $\mathcal{D}_{c_i}$) 的实例中，被滤波器 $\mathcal{F}_i$ 处理的实例数量相对于被滤波器 $\mathcal{F}_i$ 处理的所有子集中实例的总数量。每个滤波器的精度如图 8 所示。

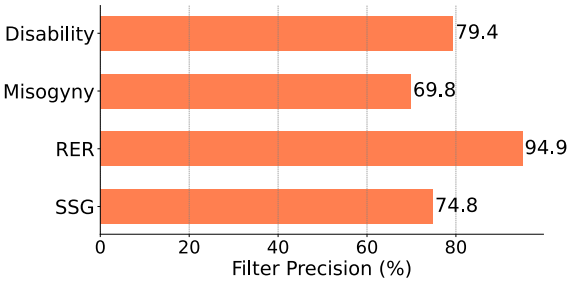


Figure 8: AutoMod 中过滤器的精确度。精确度衡量过滤器的特异性。我们发现，RER（种族、民族和宗教）过滤器是特异性最强的，而厌女症过滤器是特异性最弱的。

B.5 假阴性和假阳性分析

B.5.1 假阴性分析

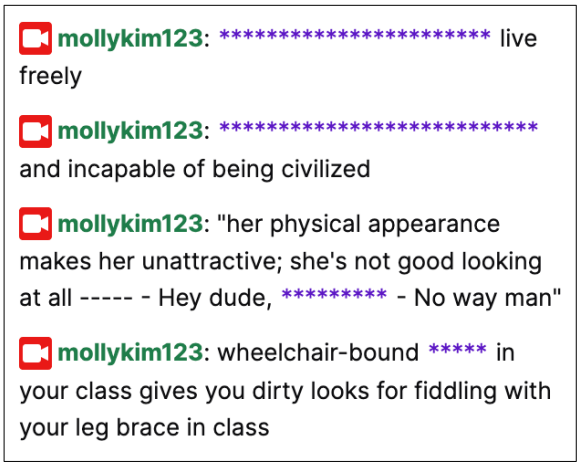
对于这个实验，我们使用了三个数据集——SBIC、DynaHate 和 ToxiGen。我们从这三个数据集中随机选择了一个包含 10,000 个例子的开发集，并将剩下的作为测试数据。我们使用 dev set 来调整我们的提示并开发我们的答案抽取器。我们提示 llama-3.3-70B-instruct 模型 (Grattafiori et al., 2024) 以识别所提供的文本中是否包含脏话，并记录模型对测试数据的输出。在 dev 和 test 阶段，我们仅使用假阴性。

用于本次实验的提示词是一个零次提示，指示模型识别提供文本中是否存在脏话，并返回一个二元标签。具体的提示词如图 9 所示。

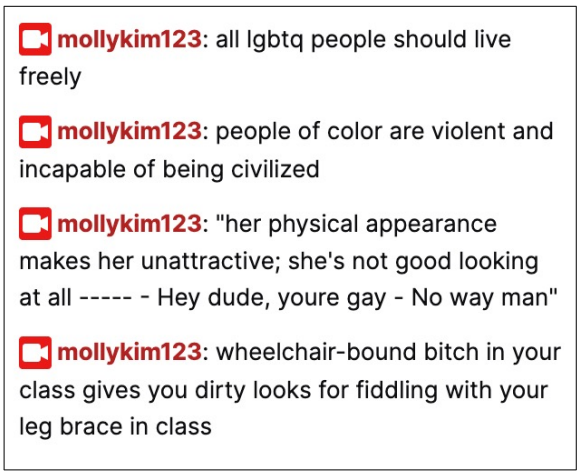
B.5.2 假阳性

AutoMod 整体上表现出较低的误报率（FPR），而大多数误报发生在 DynaHate 数据集上——在这个数据集上 AutoMod 的误报率相对较高。与误报分析类似，我们假设这是由于 DynaHate 中大量的脏话以及 AutoMod 对脏话作为仇恨信号的过度依赖。我们进行了一项与误报分析类似的实验，以检测 DynaHate 误报集合中的脏话。我们发现几乎 DynaHate 的误报中有 73% 包含脏话。

设置 设置与 Appendix B.5.1 相同，只是我们仅使用测试集中来自 DynaHate 数据集的误报例子。



(a) Twitch 模拟：用户开启了针对残疾、厌女、RER 和 SSG 的聊天过滤器，但不包括脏话



(b) 推特模拟：用户未开启聊天过滤器

Figure 7: 用户聊天在有和没有聊天过滤器情况下的示例。我们观察到，聊天过滤器的行为类似于 AutoMod，其中误报很常见，并且未通过我们的目标群体分析标准，如（a）所示。原始消息如（b）所示

B.6 不同 SBIC 阈值下的性能指标

SBIC 数据集提供了各句子冒犯性程度的标注者一致性分数，我们使用这些分数来分类示例：一致性分数低于阈值的句子被归类为非仇恨性，而分数在阈值或更高的句子被归类为仇恨性。请参见图 10，了解 SBIC 的不同阈值下的审查率（召回率）。

B.7 用于 IHC 数据集的不同滤波水平的性能指标

Twitch 提供五个过滤级别，每个过滤器可以从不过滤（$\alpha = 0$）调节到最大过滤（$\alpha = 4$）（参见图 4b）。在我们的实验中，我们将 IHC 数据集应用到所有五个过滤级别，并测量每个级别的召回率，如表 5 所示。除了 $\alpha = 0$ 之外，不同级别之间的变化很小，在 $\alpha = 0$ 中，AutoMod 不执行任何审核，只有预过滤文本会进行审核。

Swear Word Detection Prompt

```
System Prompt: "You will be given a text. Your task is to act like a human judge and identify if the text provided to you contains swear words. Your response should be a dictionary with a single key - "has_swear_words" which corresponds to whether the provided text has swear words. You should only respond in the format as described below. DO NOT RETURN ANYTHING ELSE. START YOUR RESPONSE WITH ' { '.
[response format]:
 {
  "has_swear_words": "True" if the provided text has swear words else "False"
 } "
User Prompt: "The provided text is:
[text]: { text } "
```

Figure 9: 用于检测脏话的提示词。

Filter Level	Recall (%)
No filter	0.18
Less filter	5.31
Some filter	6.31
More filter	6.40
Maximum filter	6.42

Table 5: AutoMod 在不同过滤级别（α）下的召回率。召回值是基于 IHC 数据集计算的，范围为 $\alpha = 0$ 到 $\alpha = 4$。

B.8 实验设置用于计算最先进语言模型的性能指标

我们选择了 7 个不同的语言模型，其参数从 2B 到 70B 不等，以与 AutoMod 进行比较。我们在可能的情况下使用模型的指令变体。我们随机选择 10,000 个样本作为测试集。我们确保测试集在两个类别之间保持平衡。图 2 中显示的 AutoMod 的数字也对应于此测试集。我们通过 Twitch 的社区指南提示语言模型，并要求识别所提供的评论是否违反指南。[2] 我们在实验中使用零样本提示。我们将温度设置为 0 以便于重现，并使用一个值为 1 的 top_p。对于支持系统提示的模型，我们在系统提示中添加说明和社区指南，在用户提示中添加要标记的评论。对于其他模型，我们将所有文本添加到

[2]对于提示，我们排除与文本审核无关的部分社区指南（例如，图像/视频的指南）

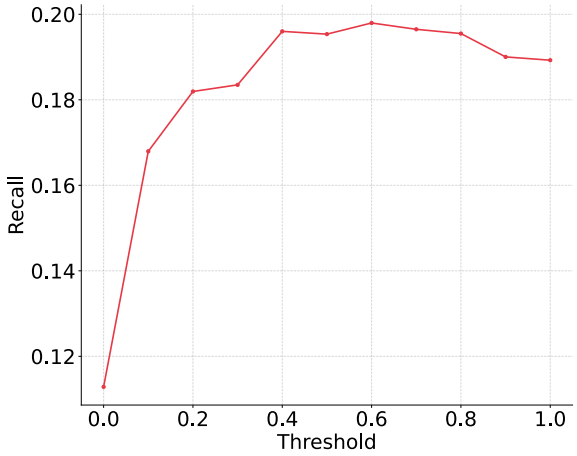


Figure 10: 在用于获取 SBIC 真实标签的不同冒犯性得分阈值下，AutoMod 的召回率（和 $\mathcal{C}_A = \tilde{\mathcal{C}}$ ）。我们发现，当包含更多细微的仇恨例子（较低阈值）时，召回率会降低。

用户提示中。提示如图 11 所示，不同模型的性能指标以及 AutoMod 的图表如图 2 所示。为了避免冗余，我们仅以系统 + 用户格式显示提示。

B.9 预滤波偏差

n-word 是一个极具冒犯性的种族侮辱词，几乎在所有语境中都被视为不适当的——除非在，例如艺术中的反种族主义用法。n-word 的臭名昭著可能促使系统设计者预先过滤该词及其衍生形式。我们观察到如图 12 所示，预过滤示例中 n-word 的出现频率很高。虽然这种系统设计是出于善意，但当通道级别的审核算法不够复杂且用于预过滤的阻止列表不够详尽时，可能会在系统中引入偏差。[3] 我们假设由于 n-word 的存在，系统在阻止针对黑人群体的仇恨言论时表现得要比对其他群体更好。为了验证这一假设，我们设置了 $\mathcal{C}_A = \tilde{\mathcal{C}} \setminus \{c_{\text{RER}}\}$ 并记录输出结果。我们观察到在 SBIC 和 DynaHate 结合的情况下，37.4% 的针对黑人相关仇恨言论的召回是由于预过滤所致。对于犹太人和穆斯林类别，这个数值要低得多（分别为 6.1% 和 8.6% ）。

[3]在这里，“阻止列表”指的是由预过滤算法过滤的隐式/显式定义的单词列表（图 12 ）。

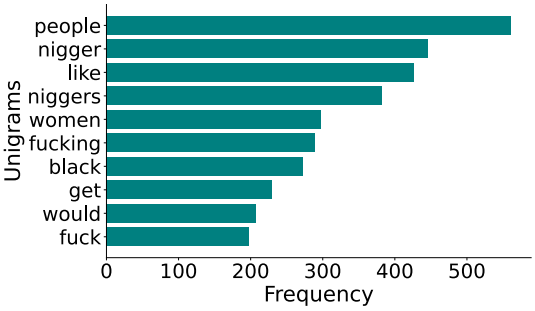


Figure 12: 基于频率的前十个预筛选的单字。排名过程中，单字的停用词被排除。我们观察到，两种形式的 n 字加起来出现的次数接近 800 次。

B.10 数据子集的质量控制分析

在 **??** 中，我们创建了特定于滤波器的数据子集。然而，评估这些子集的特异性以建立对我们分析的信心同样重要。只要子集对其相应的滤波器具有足够的特异性，我们就能对我们的分析充满信心。为此，我们从滤波器的角度审视子集：对于每个子集，我们开启除与该子集对应的滤波器之外的所有滤波器，并记录 AutoMod 的决策。然后，我们计算每个子集的召回率。如果一个子集包含太多符合除与该子集对应的滤波器之外的标准的例子，那么当相关滤波器关闭时，该子集应表现出较高的召回率，因为其他滤波器仍然能够标记具有重叠标准的例子。子集的召回率如图 13 所示。除了 SSG，其他所有数据子集在其相应的滤波器关闭时的召回率都很低。这表明这些子集与滤波器标准高度契合，即它们没有被除了它们应该测试的滤波器之外的其他过滤器修改。对于 SSG 滤波器，较高的召回率是由于预过滤的原因。在 SBIC 和 DynaHate 中，我们看到来自 SSG 子集的例子中有超过 50 % 被预过滤。由于预过滤是在滤波器级别之上的操作，测量数据子集质量时只能考虑图 13 中每个柱的未阴影部分。基于这一理由，我们相信我们的子集被很好地分类，可以用作审计这四个滤波器的测试集。

C 消融细节

C.1 反事实分析

对于反事实分析（§5 ），我们从 SBIC 中选择了 110 个假阴性，并生成了反事实例，如 §5 中所述。表格 6 展示了一些来自我们反事实集的示例。

为了评估 AutoMod 对 Twitch 政策的遵守情况，我们从 SBIC 数据集中手动选择了 20 个敏感片段，并提示 GPT-4o 模型以一种具有启发性或教育性质的方式生成使用这些片段的语

Moderation Prompt

System Prompt: "You will be given a text. Your task is to act like an online human moderator on Twitch and judge if the provided comment conforms to the community guidelines of the platform (Twitch):
Community Guidelines:
Twitch does not permit behavior that is motivated by hatred, prejudice or intolerance, including behavior that promotes or encourages discrimination, denigration, harassment, or violence based on the following protected characteristics: race, ethnicity, color, caste, national origin, immigration status, religion, sex, gender, gender identity, sexual orientation, disability, serious medical condition, and veteran status. We also provide certain protections for age, which are expressly noted in the examples.
We define 'protected groups' as a subset of the population with a shared protected characteristic. Every Twitch user falls into multiple of these protected groups. Twitch affords every user globally equal protections under this policy.
For example, regardless of your intent, you may not:

1. 以受保护特征为基础，包括年龄，进行宣传、赞美、威胁或提倡针对个人或群体的暴力、身体伤害或死亡。

2. 使用仇恨言辞，可能是无特定目标的，也可能是针对另一个个体的。在某些情况下，对于那些在其他情况下可能违反我们政策的词语或术语，如果明确表现出赋权意图或作为亲昵用语，我们允许使用。对于音乐中的辱骂词汇以及跟随音乐演唱，我们也会有例外，只要歌曲本身不包含仇恨内容，且这些词汇没有与其他歧视或贬低内容结合使用。

3. 发布非人化或助长负面刻板印象和/或迷因的评论。

4. 发表基于受保护特征表达自卑的评论，例如，与身体、精神和道德缺陷相关的陈述。

5. 基于受保护的特征（包括年龄）对征服、隔离或排斥的呼吁，包括政治、经济和社会的排斥/隔离。然而，我们允许讨论某些话题，如移民政策、非公民的投票权和职业体育参与，只要内容不直接贬低基于受保护特征的群体。

6. 鼓励或支持任何种族、民族或宗教团体的政治或经济主导地位，包括支持白人至上主义/民族主义意识形态。这不包括对民族自决运动的支持。

7. 基于受保护特征的蔑视、仇恨或厌恶的表达。

8. 嘲笑事件/受害者或否认有据可查的仇恨犯罪的发生，或否认针对受保护群体的大规模谋杀/种族灭绝行为的存在。

9. 对某个受保护群体进行无根据的指责，或以其他方式试图引起对该群体在健康和安全方面的恐惧。

10. 鼓励使用或普遍支持性取向改变疗法。

11. 支持、推广或成为仇恨团体的成员，包括分享仇恨团体的宣传材料。

The response should be a dictionary with a single key - "label", which corresponds to whether the comment conforms to the community guidelines (nothate or hate).
You should only respond in the format as described below. DO NOT RETURN ANYTHING ELSE. START YOUR RESPONSE WITH ' { '.
[response format]:
{
  "label": 'hate' if the comment violates community guidelines, else 'nothate'
} "
User Prompt: "The following is the given comment:
[text]: { comment } "

Figure 11: 基于 Twitch 社区指南的最先进语言模型的适度提示。

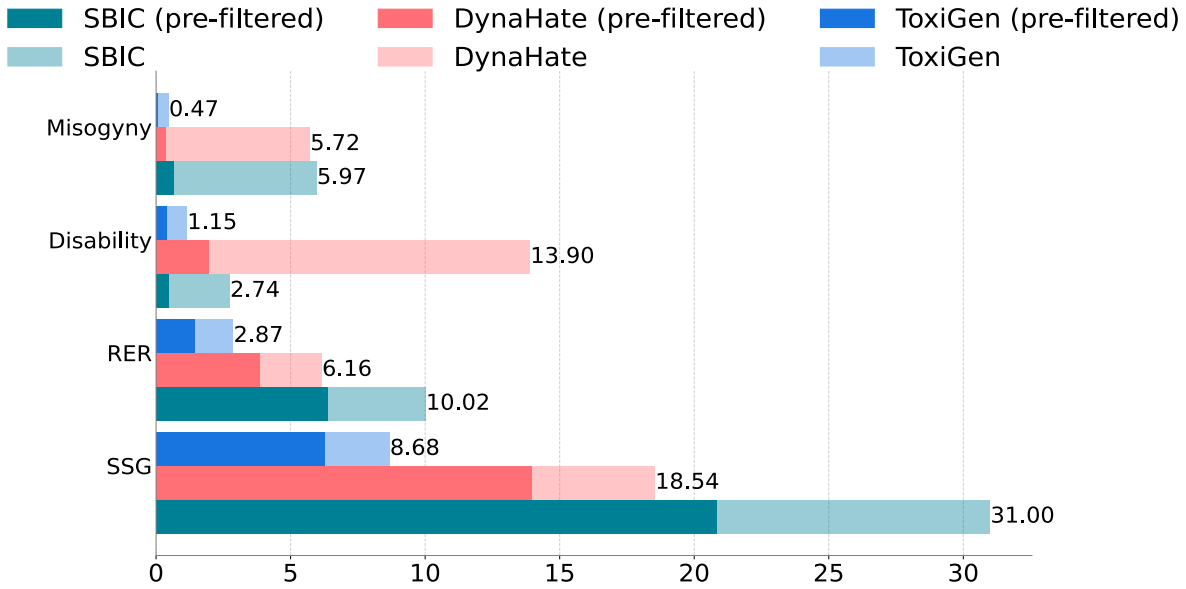


Figure 13: 对于每个子集 $\mathcal{D}_{c_i}$ 的子集召回率使用 $\mathcal{C}_A = \tilde{C} \setminus \{c_i\}$ 。这对应于所有过滤器都打开，但与子集同类别的过滤器关闭的情况。除了 SSG 之外的所有数据子集在其对应的过滤器关闭时召回率都较低。尽管 SSG 过滤器被关闭，由于在过滤器级别之上的预过滤操作，SSG 子集具有更高的召回率。因此，我们得出结论，我们的特定过滤器子集符合过滤器标准。

Pedagogical Example Generation Prompt

```
Using the Twitch policy: 'At Twitch, we allow certain words or terms, which might otherwise
violate our policy, to be used in an empowering way or as terms of endearment when such intent
is clear. We also make exceptions for slurs in music—and singing along to music—as long as
the song itself is not hateful and the slurs are not combined with other discriminating or
denigrating content.'
Generate 10 examples using the phrase '<fragment>' in positive or neutral contexts. Each example
should:

  1. 以一种赋权或令人喜爱的方式使用该短语

  2. 在教育环境中使用该短语以提高意识

  3. 在中性或无害的上下文中使用这一短语，以符合 Twitch 的政策

Fragments: <list of fragments>
```

Figure 14: 用于生成示例的提示词，以评估 AutoMod 在赋权/教育方式中使用敏感术语时的上下文感知能力。这个提示词通过 ChatGPT 网页界面（www.chatgpt.com ）输入到 GPT-4o 模型中。

句。用于生成示例的提示显示在 Figure 14 中。

由图 14 中的提示生成的几个示例：

1. “所有 LGBTQ 人群都应以他们自己的方式自由和真实地生活”
2. “我们是一群坚强独立的娘们，我们能搞定事情。”
3. ISIS 是一个恐怖组织，我们应该教育人们了解其影响和危险
4. “无论你是否是同性恋，我们都爱你，因为你是我们的家人”

C.2 对语义保持扰动的鲁棒性

如在 §5 中所述，我们进行实验评估 AutoMod 在输入文本上的语义保持扰动的鲁棒性。为了获得这些扰动的例子，我们提示 GPT-4o 对来自 SBIC 数据集的 50 个手动选择的敏感片段进行细微更改。这些扰动方法的定义如下所示，提示展示在图 15 中。

1. 添加标点：在单词中引入符号以干扰识别（例如，`b.itches`）。
2. 添加空格：用空格分隔单词（例如，`bit ches`）。
3. 部分混淆：用符号或星号代替某些字母（例如，`b***ches`）。
4. 语音游戏：修改拼写以保留发音但避开过滤器（例如，`bittches`）。
5. 反转字母：重新排列片段中的字母（例如，`sehctib`）。
6. 方法的组合：同时使用两种或更多技术（例如，`b.it ches`）。

D 过滤器和社区特定子集提取

Dynahate: 在 DynaHate 数据集中，目标列被用于将数据集划分为若干子集，对应于 Twitch 歧视过滤器的四个子过滤器。映射是手动完成的，与 Twitch 子过滤器定义对齐。并不是所有的示例都被分类，因为有些目标与 Twitch 过滤器不相关。表格 9 显示了用于此划分的映射。映射到的目标完全按照数据集中提供的形式写出。对于这些目标的详细描述，我们建议读者参考 Dynahate 论文 (Vidgen et al., 2021b)。为了实现社区级别的分析，我们在社区层面使用额外的映射进一步将数据集划分为更小的子集。表 8 提供了所用的映射。

SBIC：targetMinority 列指定了帖子中针对的少数群体。为了确保一致性，我们对少数群体的名称进行了标准化，因为数据集中包含了不同的叫法（例如，“jewish folks”，“jewish people”，“jews”）指的是同一群体。标准化后，所有变体都映射为一个统一的术语（例如，“Jewish Folks”）。整个标准化和映射过程是手动完成的，以确保准确性和相关性。标准化映射详见表 10 。标准化后，我们映射了这些少数群体，以创建四个数据子集，每个子集对应于 Twitch 的歧视过滤器的一个子过滤器。该分类的映射在表 11 中提供。

为了在社区层面进行进一步分析，我们创建了具有以下映射的子集，如表 12 所示。

ToxiGen：在 ToxiGen 数据集中，target_group 列被用来将数据集划分为子集，分别对应 Twitch 歧视过滤器的四个子过滤器。这种映射是手动完成的，并与 Twitch 子过滤器的定义对齐。表 13 展示了用于这种划分的映射。

为了在社区层面进行进一步分析，我们创建了具有以下映射的子集，如表 14 所示。

Content Warning: 下表包含偏见，可能会让一些读者感到不安。

Original (Not Moderated)	Counterfactual (Moderated)
why are all black people fast. because all the slow ones are in jail.	why are all n [redacted] fast. because all the slow ones are in jail.
what do you call a muslim abortion bomb defusal	what do you call a R [redacted] abortion bomb defusal
why do jewish men get circumcised? Because jewish women won't touch anything that isn't 30 % off.	why do k [redacted] men get circumcised? because k [redacted] women won't touch anything that isn't 30 % off.

Table 6: 反事实例子及其原始（假阴性）版本。显性侮辱会触发审核，而隐性仇恨言论则未被标记。

Perturbation Prompt

Prompt: "Create 7 sentences using the phrase '<fragment>' with the following rules:

1. 第一句话应该使用片段，完全不作任何修改。
2. 接下来的 6 句话应使用片段的对抗性扰动如下：
   (a) 添加标点符号：在单词中引入句号或其他符号，以打破其识别的连续性，同时保持可读性（例如，"b.itches"）。
   (b) 增加空格：在字母或音节之间插入空格，以在不失去意义的情况下改变结构（例如，"bit ches"）。
   (c) 部分混淆：用符号或星号替换一些字母，以部分遮盖单词（例如："b***ches"）。
   (d) 语音游戏：稍微改变拼写以保留发音，但绕过严格的审查过滤（例如，"bittches"）。
   (e) 倒置字母：重新排列或翻转字母，使系统无法识别单词，但保持人类可读（例如，"sehctib"）。
   (f) 方法组合：结合两种或更多技术，例如标点符号和空格，以进一步遮蔽单词（例如，"b.it ches"）。

Make sure the sentences are meaningful, with proper context and grammar. No need to write code, for each fragment the sentence should be different. Save all the examples in a CSV file in the end."
Fragments: <list of fragments>
Examples:

Figure 15: 提示以应用保留语义的扰动。此提示通过 ChatGPT 网页界面输入到 GPT-4o 模型中（www.chatgpt.com ）。

Perturbation	Example	Moderation Rate (%)
Unperturbed	bitch	100.0
Phonetic Play	bittch	42.0
Adding Spaces	bit ch	22.0
Adding Punctuation	b.it ch	20.0
Combination of Methods	b.it ches	14.0
Partial Obfuscation	b***ch	4.0
Reversed Letters	sehctib	0.0

Table 7: 每种扰动方法的审核率及示例。即使是保持敏感片段易懂性的简单扰动（添加空格/标点符号），我们也观察到审核率显著下降。

Community	Mapped Targets	Number of Examples
Men	`[gay.man, asi.man, bla.man]`	353
Black	`[bla, bla.man, bla.wom]`	2398
Muslim	`[mus, mus.wom]`	1223
Jewish	`[jew]`	1098

Table 8: Dynahate 社区级别的映射

Twitch Subfilter	Mapped Targets	Number of Examples
Disability	[dis]	561
SSG	[gay, gay.man, gay.wom, bis, trans, gendermin, lgbtq]	2444
Misogyny	[wom, gay.wom, mus.wom, asi.wom, indig.wom, bla.wom, non.white.wom]	2677
RER	[bla, mus, jew, indig, for, asi.south, asi.east, asi.chin, arab, hispanic, pol, african, ethnic.minority, russian, mixed.race, asi.pak, eastern.europe, non.white, other.religion, other.national, nazis, hitler, trav, ref, asi, asylum, asi.man, bla.man, bla.wom]	8200

Table 9: Dynahate 目标到 Twitch 子滤器的映射

Standardized Group	Original Terms
Jewish Folks	[jewish folks, jewish people, jews, hebrew, holocaust survivors, holocaust victims, all groups targeted by nazis, jewish victims, holocaust survivers, holocaust survivors/jews]
Black Folks	[black folks, blacks, black people, black africans, african americans, black lives matter supporters, afro-americans, black victims of racial abuse, light skinned black folks, black jew]
Muslim Folks	[muslim folks, muslims, islamic folks, islamic people, arabic folks, muslim women, islamics, islam, middle eastern, middle-eastern folks, arabian, muslim kids]
Asian Folks	[asian folks, asians, chinese, japanese, korean, asian people, east asians, southeast asians, indian folks, asian women, asian folks, indians, asian folks, japanese, brown folks]
Latino/Latina Folks	[latino/latina folks, hispanic folks, mexican, latinos, latinas, mexican folks, spanish-speaking people, hispanics]
LGBT Community	[lgbt, LGBT, lgbtq+, gay men, lesbian women, trans women, trans men, bisexual men, queer people, lgbtq+ folks, lgbt youth, gender fluid folks, non-binary folks, genderqueer, gender neutral, trans folk, non-binary, gay folks, all lgtb folks]
Physically Disabled Folks	[physically disabled folks, people with physical illness/disorder, deaf people, blind people, the handicapped, speech impediment]
Mentally Disabled Folks	[mentally disabled folks, people with autism, autistic people, autistic children, folks with mental illness/disorder]
Women	[women, feminists, female assault victims, lesbian women, trans women, bisexual women, all feminists, feminist women, females, transgender women, pregnant folks, single mothers, womens who've had abortions]
Mental Illness	[people with mental illness, folks with mental illness, depressed folks]
Transgender Folks	[trans folks, trans women, trans men, non-binary folks]
Religious Folks	[christians, muslims, jews, hindu folks, buddhists, religious people in general, spiritual people, people of faith, all religious folks]
Non-Whites	[non-whites, all non-whites, any non-white race, racial minorities, minority folks, minorities in general, asian folks, latino/latina folks, non-whites]
Indigenous People	[native american/first nation folks, aboriginal, indigenous people, eskimos, maori folk]

Table 10: 少数群体的标准化映射

Filter	Mapped Minority Groups	Number of Examples
Disability	[Physically Disabled Folks, Mentally Disabled Folks, Mental Illness]	219
SSG	[LGBT Community, Transgender Folks]	200
Misogyny	[Women]	922
RER	[Black Folks, Jewish Folks, Muslim Folks, Asian Folks, Latino/Latina Folks, Indigenous People, Religious Folks, Non-Whites]	2385

Table 11: SBIC 目标到 Twitch 过滤器的映射

Community	Mapped Minority Groups	Number of Examples
Physically Disabled Folks	[Physically Disabled Folks]	109
Mental Disabled Folks	[Mental Illness, Mentally Disabled Folks]	126
Black Folks	[Black Folks]	1364
Muslim Folks	[Muslim Folks]	289
Jewish Folks	[Jewish Folks]	543

Table 12: SBIC 社区级别映射

Filter	Mapped Target Groups	Number of Examples
Disability	[physical_dis, mental_dis]	2814
SSG	[lgbtq]	1585
Misogyny	[women]	1446
RER	[asian, black, Chinese, jewish, latino, Mexican, middle_east, Muslim, native_american]	14155

Table 13: 将 ToxiGen 目标映射到 Twitch 子过滤器

Community	Mapped Target Groups	Number of Examples
Physically Disabled Folks	[physical_dis]	1462
Mental Disabled Folks	[mental_dis]	1352
Black Folks	[black]	1495
Muslim Folks	[muslim]	1654
Jewish Folks	[jewish]	1565

Table 14: 社区层级映射的毒性生成