

网规模训练数据集的投毒攻击是可行的

Nicholas Carlini¹ Matthew Jagielski¹ Christopher A. Choquette-Choo¹ Daniel Paleka²
Will Pearce³ Hyrum Anderson⁴ Andreas Terzis¹ Kurt Thomas¹ Florian Tramèr²

¹Google

²苏黎世联邦理工学院

³NVIDIA

⁴强大的智能

摘要

深度学习模型通常在从互联网上爬取的分布式、网规模数据集上进行训练。在本文中，我们介绍了两种新的数据集投毒攻击，这些攻击有意地向模型的性能中引入恶意示例。我们的攻击立即可行，今天就可以对10个流行的数据集进行投毒。我们的第一种攻击，分割视图投毒，利用了互联网内容的可变性，确保数据集注释者对数据集的初始视图与后续客户端下载的视图不同。通过利用特定的无效信任假设，我们展示了如何仅以60美元的价格就可以对LAION-400M或COYO-700M数据集中的0.01%进行投毒。我们的第二种攻击，前置投毒，针对定期快照众包内容的网规模数据集，例如维基百科，攻击者只需要一个有限的时间窗口来注入恶意示例。鉴于这两种攻击，我们通知了每个受影响数据集的维护者，并推荐了几种低开销的防御措施。

1 引言

用于训练深度学习模型的数据集已经从数千个经过精心策划的示例 [20, 33, 41] 增长到从互联网自动爬取的数十亿个样本的网规模数据集 [10, 48, 53, 57]。在这个规模下，手动策划和确保每个示例的质量是不可行的。

迄今为止，这种数量优先于质量的权衡被认为是可以接受的，这是因为现代神经网络对大量标签噪声 [55, 83] 非常有韧性，并且在嘈杂数据上进行训练甚至可以提高模型在分布外数据上的效用 [50, 51]。

尽管大型深度学习模型对随机噪声有很强的韧性，但是训练集中即使是微小的对抗噪声（即投毒攻击 [6]）也足以在模型行为中引入有针对性的错误 [14, 15, 60, 76]。这些之前的研究认为，由于缺乏人工策划，现代深度学习模型受到投毒攻击的威胁是实际存在的。然而，尽管存在潜在的威胁，据我们所知，目前还没有

涉及网规模数据集投毒的现实攻击已经发生。一个解释是之前的研究忽略了一个问题，即对手如何确保他们的损坏数据被纳入网规模数据集中。

在本文中，我们介绍了两种新颖的投毒攻击，可以确保恶意示例出现在用于训练当今最大的生产机器学习模型的网规模数据集中。我们的攻击利用了网规模数据集当前信任假设的关键弱点：由于货币、隐私和法律限制的结合，许多现有数据集不作为静态、独立的文件发布。相反，数据集要么由客户端必须爬取的网页内容索引组成，要么由客户端下载的定期网页内容快照组成。这使得攻击者可以确定要投毒的网页内容（并且，正如我们将展示的那样，甚至可以确定何时投毒这些内容）。

我们的两种攻击的工作原理如下：

- 分割视图数据投毒：我们的第一种攻击针对当前的大型数据集（例如LAION-400M），并利用了数据集维护者在收集时所看到的数据可能与训练时的最终用户所看到的数据不同（显著且任意）。这种攻击是可行的，因为缺乏（加密的）完整性保护：无法保证客户端在爬取页面时观察到的数据与数据集维护者将其添加到索引时观察到的数据相同。

- 前置数据投毒：我们的第二种攻击利用了包含周期性用户生成内容快照的流行数据集，例如维基百科快照。在这种情况下，如果攻击者能够在快照之前精确地定时进行恶意修改，以便将其包含在网规模数据集中，他们就可以预先收集数据。这种攻击是可行的，因为快照时间表是可预测的，内容审核存在延迟，并且快照是不可变的：即使内容审核员在事后检测到并恢复了恶意修改，攻击者的恶意内容仍将在用于训练深度学习模型的快照中存在。

我们在10个流行的网规模数据集上探索了这两种攻击的可行性。我们展示了这些攻击是实际可行的，即使对于资源有限的攻击者也是如此：仅需60美元，我们就可以在2022年投毒0.01%的LAION-400M或COYO-700M数据集。

为了对抗这些攻击，我们提出了两种防御措施：

- 完整性验证通过分发所有索引内容的加密哈希来防止分割视图投毒，从而确保客户端观察到与维护者首次索引和注释时相同的数据。
- 基于时间的防御措施通过随机化数据快照的顺序或延迟内容的包含并应用来自可信任调解人的还原来防止前置投毒。

我们讨论了这些防御措施的局限性（例如，在完整性检查的情况下，防止良性修改，如重新编码、重新调整大小或裁剪图像）以及更强大、面向未来的解决方案，其中信任假设更少。

负责任的披露。我们向本研究中出现的10个数据集的主要维护者披露了我们的结果。其中六个数据集现在采用了我们推荐的完整性检查实施。此外，我们还提供了补丁，以支持最流行的网规模数据集下载器进行完整性检查。最后，我们已经通知维基百科关于他们数据收集过程中的前置投毒漏洞。

2 背景和相关工作

我们的工作基于对投毒网规模数据集风险的现有知识，但重点放在实际利用这种攻击的攻击向量上。我们概述了为什么网规模数据集变得至关重要，已知的网规模数据集安全风险，以及由于将未经筛选的数据纳入模型中而产生的辅助数据集质量问题。

朝着未经筛选的数据集的势头。当应用于大规模数据集时，深度学习最为有效[10, 30]。但是，筛选这样的数据集是昂贵的。即使没有手动标记，训练数据的可用性也已成为进一步提高模型效用的限制因素。例如，最近的Chinchilla [27]语言模型中观察到的缩放定律表明，训练一个计算最优的5000亿参数模型需要110万亿个训练数据标记-比目前用于训练这种规模模型的数据多10倍[19]。为了大幅扩大数据集的规模，从越来越广泛的不受信任和未经筛选的网络来源中抓取数据已经变得很常见。

毒害攻击的安全风险。未经筛选的训练数据集是投毒攻击的主要目标[6]。例如，攻击者可以修改训练数据集（“毒害”它），以便某些目标示例被错误分类

由于在该数据集上训练的模型。早期的毒害攻击旨在针对完全监督的分类器[18,24,64]，这些分类器是在筛选过的数据集上训练的。这些攻击通常旨在“隐蔽”，通过以对人类注释者不可察觉的方式改变数据点[74]。对未经筛选的数据集的攻击不需要这种强属性。最近的研究[14,15]表明，仅仅投毒未经筛选的网规模训练数据集的0.001%就足以引发目标模型的错误，或者植入模型的“后门”[18,24]。

因此，已知如果攻击者能够在网规模数据集的一部分中投毒，那么他们可以造成重大伤害。然而，目前尚不清楚攻击者如何在任何常见的训练数据集中放置他们的毒害样本，而不事先猜测哪些部分将被收集。本文回答了这个问题。

与数据质量相关的辅助风险。花费时间和精力来策划数据集除了安全性之外还有好处。可能最重要的是，未经策划的数据对公平性、偏见和伦理有严重影响[8, 50, 77]。例如，Birhane等人[7]指出，LAION-400M存在“令人困扰和明确的强奸、色情、恶意刻板印象、种族主义和种族歧视以及其他极具问题的内容”。许多语言数据集也包含类似有害的“仇恨言论和性暗示内容，即使经过过滤”[40]。

数据集策划并不能完全解决这些问题。

Birhane等人指出，“没有仔细的上下文分析，过滤机制很可能会审查和抹去边缘化的经历”。任何选择性地删除某些数据源而不是其他数据源的过滤方法都应该仔细考虑这样做的安全性问题，以及其他更一般的数据质量指标。

3 威胁模型和攻击场景

在介绍我们攻击的实现细节之前，我们先介绍关键术语、我们的威胁模型以及我们两种攻击背后的直观描述。

3.1 术语

由于以独立的方式分发网规模数据集是不可行的（由于数据集的大小或监管问题），当前的训练数据集可以分为两类。在第一类中，维护者生成一组 N 个样本 $\{(url_i, c_{-i})\}$ ，其中包括资源标识符 url_i 和辅助数据 c_{-i} （通常是一个标签）。我们用 t_i 表示第 i 个样本最初收集的时间。关键是，维护者不提供与 url_i 相关的数据的快照，这要么是因为存储成本不可行[3,4,17,31]，要么是因为隐私问题[13,72]，要么是因为版权限制[16]。因此，我们将这些称为分布式数据集。一个例子是LAION-5B数据集[57]，它包含五十亿个元组。

图像URL和相应的文本标题-对应于数百TB的数据。

在第二类数据集中，策展人生成一个数据集的快照 $\{x_i\}_{i=1}^{N^i}$ ，其中每个样本 x_i 从一组URL $\{url_i\}_{i=1}^{N^i}$ 在时间 t_i 抽取，然后公开提供此快照。由于这些URL提供的数据随时间变化，策展人经常（例如，每月）重新收集数据集快照，以便用户获得最新的数据视图。我们将其称为集中式数据集。例如，维基百科和Common Crawl定期生成其整个数据库的快照。这简化了训练大型模型的人们的访问，同时也阻止研究人员直接重新抓取数据库。

一旦这两种类型的数据集被发布，一个客户端（例如研究人员或应用实践者）会下载训练数据集 \mathcal{D} 的本地副本，通过在将来的时间 $t'_i > t_i$ 爬取分散式数据集 $\{(url_i, c_i)\}_{i=1}^{N^i}$ ，或者下载集中式数据集 $\{x_i\}_{i=1}^{N^i}$ 。实际上，客户端通常使用由第三方开发和维护的下载工具。

3.2 威胁模型

我们假设存在一个相对不熟练、资源有限的对手，可以在某个时间点篡改少量URL $\{url_i\}_{i=1}^{N^i}$ 的内容，使得当客户端或策展人在将来的某个时间 t_i 访问资源 i 时，他们会收到一个修改过的（投毒的）数据集 $\mathcal{D}' = \mathcal{D}$ 。投毒数据集和预期数据集之间的差异必须足够大，以至于在 \mathcal{D}' 上训练的模型 f 对某些期望的输入会产生投毒结果。

我们将 $S_{adv} \subset \{url_i\}_{i=1}^{N^i}$ 定义为对手可以修改的URL集合。

我们假设对手对策展人、下载者或维护者没有专门或内部知识，除了知道用于生成 \mathcal{D} 的URL集合 $\{url_i\}_{i=1}^{N^i}$ （这些信息由数据集维护者或策展人发布）。我们进一步假设所有的维护者、策展人和下载者都诚实行事，不以任何方式帮助对手。因此，对手对辅助数据 c_i （例如，监督标签或文本描述）没有控制权，也不能添加或删除任何将被客户端或策展人爬取的训练数据中的URL。

我们做出了两个关键（但现实的）假设，这使得我们的攻击成为可能。对于分布式数据集，我们假设客户端不会将下载的本地数据集副本 \mathcal{D}' 与维护者索引的原始数据集 \mathcal{D} 的加密完整性进行比较。对于集中式数据集，我们假设至少需要一些时间 Δ ，才能检测到数据集中任何URL url_i 上托管内容的恶意更改（例如，对于维基百科， Δ 是恢复恶意编辑所需的时间）。因此，如果攻击者投毒了内容，策展人无法检测到 url_i 托管的毒害内容。

在数据集快照中包含 url_i 的内容的时间为 t_i ，其中任意时间 t'_i 满足 $t - \Delta \leq t'_i \leq t_i$ 。正如我们将要展示的，这些假设对于几乎所有现代网规模数据集都成立。我们在第6节中讨论了如何通过加密完整性检查和随机爬取来减轻我们所描述的攻击。

3.3 攻击场景

我们提出了两种攻击策略，用于污染最近的网规模数据集。在接下来的章节中，我们将展示这些攻击在实际的真实数据集上的有效性，并描述我们遵循的道德保障措施以最小化伤害。我们的攻击重点是我们对数据集污染研究中独特的机制。其他潜在的安全漏洞（例如，对手能够干扰客户端的未加密网络请求，或者利用网站漏洞注入新内容）超出了我们的范围，并且只会提高我们的攻击成功率。

分割视图污染。我们的第一种攻击利用了一个事实，即由主要维护者发布的分布式数据集的索引无法修改，但数据集中的URL内容可以修改。¹ 这使得能够对数据集中索引的网络资源施加持续控制的对手可以污染最终用户收集的数据集。

我们攻击中利用的特定漏洞是从一个相当简单的观察结果中得出的：仅仅因为在数据集最初收集时，一个网页托管了良性内容，这并不意味着同一个网页当前托管的是良性内容。特别是，域名偶尔会过期，当它们过期时，任何人都可以购买它们。我们展示了在大型数据集中，域名过期是异常常见的。

攻击者不需要知道客户端将来下载资源的确切时间：通过拥有该域名，攻击者保证任何将来的下载都会收集到被投毒的数据。

我们注意到攻击者已经经常购买过期的域名来劫持与这些域名相关的残留信任[35, 39, 67]。攻击者过去曾针对残留信任攻击已停用的银行域名[44]和导入的JavaScript库[47]，以提供恶意软件或窃取用户数据，接管与该域名相关联的电子邮件地址[56]，控制授权名称服务器[39]，或仅仅提供广告[36]。在这里，我们滥用残留信任来污染分布式数据集。虽然更复杂的攻击可能会实现相同的目标，例如利用网站、迫使网站所有者修改内容或修改未加密的网络流量，但我们在这项工作中专注于域名过期这一自然现象。

为了选择要购买的域，攻击者可以优先选择在数据集中托管多个URL的廉价域

¹ 换句话说，C类型“int const *”（从业者通常如何处理这些基于URL的数据集）和“int * const”（数据集实际提供的内容）之间存在重要区别。

(最小化每个投毒URL的成本)，或者托管具有特定辅助数据的内容的域（请记住，攻击者无法修改分布式数据集索引中包含的辅助数据）。我们证明了分割视图投毒在实践中是有效的，因为大多数网规模数据集的索引在首次发布后长时间保持不变，即使其中一部分数据过时。而且，很少有（甚至没有现代的）数据集包含任何形式的下载内容的加密完整性检查。

前置投毒。我们的第二种攻击将分割视图投毒的范围扩展到攻击者无法持续控制数据集索引的网络资源的设置。相反，攻击者只能在恶意修改被检测到之前的短时间内（例如几分钟）修改网页内容。

这种设置在聚合众包网页上发布的内容的数据集中很常见，比如维基百科上的内容。事实上，临时编辑维基百科以破坏其内容[63, 69]是很容易的。一个天真的对手可能会在任意时间污染一些维基百科内容，并希望数据集的策划者在恶意编辑被还原之前抓取到被污染的页面。然而，维基百科的破坏行为平均在几分钟内就会被还原[80]，因此任意时间的恶意编辑不太可能影响数据集的收集。

我们的前置攻击依赖于一个事实，即攻击者在某些情况下可以准确预测一个网络资源被访问并包含在数据集快照中的时间。因此，攻击者可以在策划者收集快照之前污染数据集内容，从而超前地处理稍后会还原恶意编辑的内容。我们将展示前置攻击对维基百科数据集特别有效，因为官方的维基百科快照过程按照可预测且具有良好文档记录的线性顺序访问文章。因此，攻击者可以准确预测任何维基百科文章的快照时间 t_{-i} ，精确到分钟。

4 分割视图数据投毒

我们现在开始评估，首先是分割视图数据投毒攻击，攻击者通过购买和修改过期的域名来投毒分布式数据集。

4.1 我们的攻击：购买过期域名

虽然分割视图投毒适用于任何分布式数据集，我们专注于多模态图像-文本数据集。在这种数据集中，每个URL指向由某个数据提供者托管的图像，并且辅助数据包含图像的文本描述，例如，来自网页的注释类标签或标题。

²https://en.wikipedia.org/w/index.php?title=Wikipedia:Database_download&oldid=1138465486

我们的攻击利用了域名系统(DNS)不会将任何域名永久分配给特定的个人或组织，而是授予短期(每年)的“租约”，必须经常续订。因此，域名会随着时间的推移不断过期-无论是有意还是无意-当重新注册费用未支付时。

当托管图像的域名在分布式数据集中过期时，任何人都可以支付费用来获取该域名的所有权，并在受害者客户端后续下载索引图像时返回任意内容。分割视图投毒滥用了过期域名中固有的剩余信任，就像传统的域名劫持攻击一样。我们发现，在许多流行的分布式数据集中，域名的质量保证措施相对较松散（如果有的话），因此几个月前过期的普通域名可以自由获取，以控制整个数据集的一小部分。

在本节中，我们研究了通过购买过期域名来投毒数据集的可能性。我们首先量化了流行的分布式数据集中过期的域名的比例（§ 4.2），然后测量了这些数据集被爬取的频率（§ 4.3），验证了目前尚未有人利用这种攻击（§ 4.4），最后研究了攻击的潜在下游影响（§ 4.5）。

4.2 量化攻击面

表1列出了本文研究的十个最近的数据集，它们容易受到分割视图投毒攻击。最早的三个数据集（PubFig、FaceScrub和VGG Face）是人脸数据集，并将每个图像与单个人的身份（通常是知名名人）关联起来。其余七个数据集是多模态数据集，包含指向图像的URL以及从网页的HTML中自动提取的文本标题。因此，对于这些数据集，图像可以由相应域的所有者进行修改，但图像的标题在数据集索引中是固定的。

为了衡量可能被投毒的图像比例，我们计算托管在已过期且可购买的域上的图像数量。如果该域的DNS记录不存在，我们称该域已过期。为了衡量这一点，我们在2022年5月和2022年8月在两个地理上独立的数据中心对数据集中的每个域名执行了nslookup，并且如果所有四个查询结果都是NXDOMAIN响应，则报告该域名已过期。

我们进一步称一个域名为可购买的，如果它已过期，并且至少有一个域名注册商在2022年8月明确将该域名列出出售。表1报告的是数据集中来自可购买域名的图像比例，而不是计算可购买数据的总比例（这将代表一个财务上无限制的对手）。

有些注册商将域名列为出售，即使它们实际上是由投机者拥有的。我们将这些域名从可购买域名集合中排除，因为购买这些域名通常是一项昂贵且耗时的过程。

数据集名称	大小发布加密			来自数据的 每月10000美元购买过期域名	可购买数据下载	
	($\times 10^6$)	日期	哈希?		购买过期域名	
LAION-2B-en [57]	2323	2022	X [†]	0.29%	$\geq 0.02\%$	≥ 7
LAION-2B-multi [57]	2266	2022	X [†]	0.55%	$\geq 0.03\%$	≥ 4
LAION-1B-nolang [57]	1272	2022	X [†]	0.37%	$\geq 0.03\%$	≥ 2
COYO-700M [11]	747	2022	X [‡]	1.51%	$\geq 0.15\%$	≥ 5
LAION-400M [58]	408	2021	X	0.71%	$\geq 0.06\%$	≥ 10
概念12M [16]	12	2021	X	1.19%	$\geq 0.15\%$	≥ 33
CC-3M [65]	3	2018	X	1.04%	$\geq 0.11\%$	≥ 29
VGG人脸 [49]	2.6	2015	X	3.70%	$\geq 0.23\%$	≥ 3
FaceScrub [46]	0.10	2014	✓ [§]	4.51%	$\geq 0.79\%$	≥ 7
PubFig [34]	0.06	2010	✓ ^{§*}	6.48%	$\geq 0.48\%$	≥ 15

表1：所有最近发布的大型数据集都容易受到分割视图投毒攻击。我们已向受影响数据集的维护者披露了这个漏洞。所有数据集都有超过先前工作[14]所需的投毒阈值的0.01%以上的可购买数据（在2022年）。在我们披露之前，这些数据集中的每个下载都存在漏洞。

[†] LAION-5B发布了一个“合并”版本的数据集，其中对URL和标题的文本进行了加密哈希（而不是URL的内容），因此不能保护实际图像的完整性。

[‡] COYO-700M图像使用pHash [32]进行分发，该方法可以验证良性图像的更改，但对对抗性攻击不具备鲁棒性 [29]。

[§] FaceScrub和PubFig包含加密哈希值，但数据集维护者没有提供官方下载客户端来验证这些哈希值。我们发现几乎所有这些数据集的第三方下载器都忽略了哈希值。

^{*} PubFig最初发布时没有哈希值，但在数据集的1.2版本中后来添加了哈希值。

总共花费了10,000美元购买。（图1显示了可以购买的数据集比例与总成本的关系。）为了计算这个比例，我们按照“每美元图像数”递减的顺序对域进行排序：域主机上的图像数量除以购买该域的成本。

总体而言，我们发现一个预算适中的对手可以控制我们研究的10个数据集中至少0.02%-0.79%的图像。这足以对未经筛选的数据集发动现有的投毒攻击，这些攻击通常只需要对数据的0.01%进行投毒[15]。⁴我们还发现数据集的年龄与投毒的难易程度之间存在直接关系。

较旧的数据集更有可能包含已过期的域名，因此对手可以购买更大比例的数据集。

数据集从一开始就容易受到攻击。我们上述分析的一个限制是我们在2022年8月测量了受到投毒攻击的数据集的比例，但是其中许多数据集是几年前构建的。因此，许多使用这些数据集的人很可能已经在早期日期下载了它们，因此可能不容易受到我们的投毒攻击（尽管我们将在下一节中展示，这些数据集的新下载仍然频繁发生）。幸运的是，在2022年8月30日，COYO-700M数据集在我们撰写这篇研究论文时发布。

Carlini & Terzis [15]假设对手可以修改图像和它们的标题，而我们的对手只能控制图像。在§^{4.5}中，我们将展示修改标题对于成功的投毒攻击是不必要的。

我们的投毒攻击对其造成了漏洞，并发现已经有0.15%的图像托管在已过期的域名上，这些域名的购买成本低于10,000美元。之所以在发布时过期域名的数量不为零，是因为构建这些大型数据集是一项耗时且昂贵的过程。尽管COYO-700M指数在2022年8月发布，但在收集数据集[11]之前，花费了将近一年的时间，这给最早被抓取的域名在数据集发布之前就已经过期提供了充足的时间。

衡量攻击成本。最直接的问题是，这种攻击是否可以在实践中实现。我们攻击的主要限制是购买域名的货币成本，我们使用2022年8月GoogleDomains报告的成本进行衡量。在图1中，我们展示了攻击者可以控制的数据集中图像的比例，作为他们的预算函数。我们发现，每个数据集中至少有0.01%的图像可以在每年不到60美元的预算下进行控制。

4.3 测量攻击影响

我们的攻击是“追溯性”的，意味着我们可以在数据集由策展人初始构建之后对其进行投毒，但影响仅限于在我们接管域名后下载它的人。考虑到这些数据集的初始构建已经过去了几年，很明显不会有人通过抓取URL来下载它们，而是重新使用先前下载的数据集版本。因此，为了衡量分割视图投毒攻击的潜在影响，有必要进行测量。

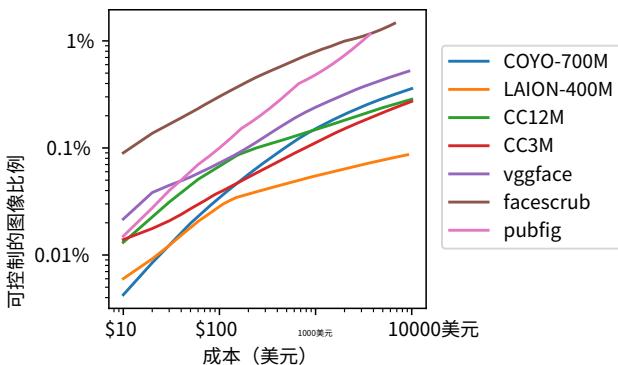


图1：控制至少0.01%的数据通常只需≤60美元。成本是按照每张图像的最低成本顺序购买域名来衡量的。

研究这些数据集实际上仍然被研究人员和从业者积极下载的速率。

方法论。我们通过购买每个列出的10个数据集中的多个过期域名，并被动地监控请求来测量从互联网上下载这些分布式数据集的速率，以测量与分布式数据集中的图像对应的URL被下载的速率。

对于每个数据集，我们购买了三个最受欢迎的过期可购买域名（即托管最多图像的可购买域名），以及三个随机选择的可购买域名。我们编写了一个简单的服务器来记录所有传入的HTTP和HTTPS⁵请求，包括访问时间、IP地址的哈希值、被请求的完整URL和任何附加的标头。这使我们能够计算这些数据集仍然被下载的频率。我们从2022年8月开始运行这个服务器，持续六个月。

分析方法。在我们的研究中，我们的服务器每月收到大约1500万个请求，平均每秒6个请求。然而，从这里开始，有必要将实际上意图下载这些数据集之一的请求与来自其他互联网用户或网络爬虫的请求分开。

为了开始我们的分析，我们做出一个（重要的）简化假设，即任何下载这些数据集之一的人都来自同一个IP地址。因此，我们可能低估了每个数据集被下载的真实速率。然后，我们说一个特定的IP地址X在时间 $[T_0, T_1]$ 内下载了数据集D，如果我们满足精确度和召回率的要求：

- 召回率：在时间范围 $[T_0, T_1]$ 内，IP地址X至少下载了数据集D中90%的URL

⁵为了监控HTTPS流量，我们从Let's Encrypt获得了证书为我们的每个域名。

在我们的控制下，包括每个域名中的至少一个URL我们拥有的6个域名的数据集。

2. 精确度：X在此时间范围内发出的请求中至少有50%是对我们控制的域名中的URL的请求 D。

这些条件是保守的，但确保我们过滤掉网络爬虫和其他大规模互联网爬取器，因为这些很可能爬取该域名的其他URL（违反了精确度约束）或者不爬取数据集中的大部分URL（违反了召回约束）。此外，因为我们每个数据集拥有六个域名，所以通过纯粹的随机机会，一个特定的IP请求来自这六个无关的域名中的URL是非常不可能的。（事实上，我们发现即使检查这三个URL也会得到几乎相同的精确度。）作为参考，对于CC-3M数据集，我们每月收到51,000个图像请求来自总共2401个唯一的IP。仅应用精确度约束，我们将其减少到2007个唯一的IP和43,000个图像请求；仅应用召回约束，我们得到70个唯一的IP和32,000个图像请求；两者结合进一步减少了64个唯一的IP和28,000个图像请求（每月）。

4.3.1 结果

我们在表1的最右列中报告了我们的结果。即使是最古老且访问频率最低的数据集，每个月至少也有3次下载。因此，在我们跟踪数据的六个月内，我们可以通过攻击来污染超过800次下载。毫不奇怪，我们发现新的数据集比旧的数据集更常被请求。

因此，不同的数据集对攻击者来说提供了不同的权衡：新的数据集具有较小比例的可购买图像，但攻击可以影响更多的易受攻击的客户端。

我们观察到，最大的十亿级图像数据集的下载频率明显低于较新的较小数据集。我们发现这是因为这些数据集很少被完整下载；相反，它们作为较小子集的上游来源。例如，公共多模态数据集（PMD）[66]和LAION-Aesthetics [59]数据集几乎完全由LAION-2B-en中的图像组成。这样的子数据集解释了为什么有时我们会看到具有非常高精度但低召回率的IP地址。

可视化数据集爬虫。通过使用我们的日志文件，我们可以将数据集下载器访问这些数据集的URL请求绘制成时间函数图，从而实现可视化。我们根据原始数据集索引的顺序对每个数据集购买的URL集合进行排序。通过这种方式，线性处理数据集索引的爬虫应该在我们的URL请求随时间变化的图中呈现出线性线条（大致上）。为了提高清晰度，我们为访问我们服务器的每个唯一IP分配一个单独的随机颜色。

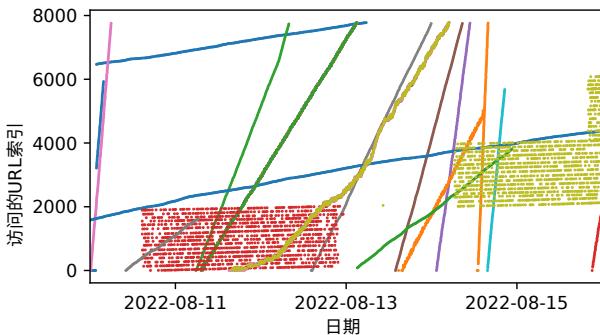


图2：用户下载概念12M的可视化。通过监控从我们购买的域名请求的URL，我们绘制每次URL请求的时间，按源IP进行彩色编码，并可以直接读取数十个用户爬取概念12M的信息。附录图8比较了各种过滤方法。

我们在图-2中展示了Conceptual 12M数据集的这个图。我们可以在这个数据中找到几个趋势。首先，大多数下载这个数据集的用户都是按照线性顺序从第一个URL到最后一个URL进行下载。然而，访问URL的速率是非常变化的：有些下载者在几个小时内爬取整个数据集，而其他人则需要几个星期才能下载完全相同的数据集。我们还观察到一些用户将数据分成块并平行下载每个块，还有一些用户暂停下载片刻，然后几个小时后恢复下载（使用不同的IP地址）。虽然我们严格的精确度和召回率要求已经给出了我们记录的IP地址确实

在下载数据集的强有力证据，但这些地址按照URL请求的线性顺序几乎可以确认这一点。实际上，由于数据集索引中URL的排序是随机的（而不是按字母顺序），数据集的下载似乎是解释为什么URL会按照特定顺序线性访问的唯一解释。

用户代理验证。最受欢迎的用户代理⁶负责我们域名^{77%}的流量。这个用户代理是硬编码的⁷在72—在2020年2月被取代—很有可能大部分请求确实来自

伦理考虑。我们实际上没有对任何数据集进行投毒。对于所有拥有的所有URL，我们返回404 Not Found响应；因此从数据集下载器的角度来看，我们购买的域名是完全透明的。我们进一步放置了一个 robots.txt 文件来防止典型的网络爬虫

从我们的域名中爬取数据集-这不太可能影响数据集的下载，因为数据集爬虫会忽略这个文件。最后，对根域名的请求返回403 Forbidden，并附带一个响应体解释该域名是研究的一部分。在这个响应中，我们列出了一个联系邮箱地址，并提供将该域名归还给原始所有者的选项，以防意外过期。我们没有收到任何关于这个地址的联系。附录E包含我们登陆网页的文本。

我们的数据收集方式是最小侵入性的。在搜索过期域名时，我们将DNS请求限制在每秒500次，我们只询问购买每个数据集中前10000个域名的成本，并最终只购买了六个。这项研究被ETH Zurich IRB认定为“豁免”。Google没有IRB，但该研究计划经过Google的专家在伦理学、人类主体研究、政策、法律、安全、隐私和反滥用等领域进行了审查。

4.4 这种攻击在野外被利用了吗？

考虑到这种攻击向量已经存在多年，并且很容易执行，不难想象过去可能有人进行过这种投毒攻击。然而，我们找不到任何这方面的证据。

我们通过寻找（1）托管的图像自初始数据集发布以来是否已被修改，以及（2）自初始数据集发布以来是否已更改所有权来寻找域购买攻击的特征。为了检测图像的变化，我们可以检查图像是否与原始图像在感知上相似（例如，使用CLIP嵌入[52]）或者是否完全相同（例如，使用加密哈希）。使用加密哈希比较图像没有误报（即，任何变化都会被检测到），但对于“良性”变化会产生误报，例如，如果一个域重新编码或调整其图像。相比之下，感知哈希的误报较少，但如果攻击者购买一个域并修改图像而保留感知哈希（这不具有冲突抗性），则可能产生误报。最后，为了检测域是否更改所有权，我们请求 whois 记录并检查最后购买日期。

结果。我们对CC3M进行了初步分析，这是一个数据集，我们拥有原始原始图像作为基准。

在所有托管超过10张图片的领域中，我们发现只有一个领域在与感知图像相似性进行比较时具有我们的攻击特征。经过进一步调查，我们发现该域名已被域名抢注者购买，并且对该域名上的图像文件的任何请求都会返回广告。如果我们比较图像的加密哈希值，我们会发现相同的域名抢注者，还有两个已被重新购买的域名。然而，进一步调查发现这些域名的所有权并未发生变化，DNS记录仅仅过期了，并且图像已被重新编码。

⁶Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:72.0)
Gecko/20100101 Firefox/72.0

⁷<https://github.com/rom1504/img2dataset/blob/fc3fb2e/img2dataset/downloader.py#L41>

我们还对LAION-400M进行了相同的分析。在这里，我们研究了数据的三个版本：原始发布版本（2021年11月）以及我们在两个后续日期（2022年04月和2022年08月）的下载版本。我们在LAION-400M中没有找到具有这种特征的域名，因此目前没有证据表明这种攻击已针对该数据集进行利用。因为我们只有该数据集的原始CLIP嵌入（原始原始字节未保存），所以我们只进行了这种比较。我们发现通过第一和第二个快照，分别有4.1M和4.2M个唯一的域名（共5.6M个）托管至少一张修改后的图像，总共找到了175M和183M个修改后的图像。我们使用CLIP余弦相似度<0.99进行了测量。我们随机抽样了几个域名，包括具有最多修改图像的700个域名。我们发现许多情况下，域名仍然由原始所有者拥有，目前正在出售，或者已被删除，但没有出现恶意行为。

4.5 将所有内容整合在一起

之前的工作[15]已经成功地对300万张图像的多模态模型进行了投毒，假设攻击者可以任意控制操纵图像的标签。然而，在本文中，我们研究的数据集超过 $100\times$ 更大，并假设攻击者无法控制文本标题。这两个变化下的投毒攻击是否有效？

我们发现它们是有效的。我们考虑两种投毒攻击目标：(1)使特定图像被错误分类为ImageNet中的某个目标标签，(2)使特定图像被Stable Diffusion Safety Filter [54]分类为NSFW。对于这些攻击目标，我们首先在LAION 400M中确定适当的文本标题，然后购买相应图像领域的图片，总共花费1000美元。然后，我们将这些图片替换为投毒样本，以模拟攻击的效果，而不会对他人造成任何伤害。

具体来说，我们使用一个OpenCLIP [28]模型，在ViT-B-32架构下进行32个时期的训练，在16个A100 GPU上批量大小为3072。对于我们的目标误分类目标，我们选择了在多个图像文本标题中出现的十个ImageNet类别，我们可以控制。当我们投毒1,000张图像（占总数据集的0.00025%）时，我们的攻击成功率为60%，可以翻转目标图像的模型零样本分类。对于我们的NSFW目标，我们在LAION 400M数据集索引中找到与（可购买域）标记为不安全的图像对应的标题。再次在1,000个投毒样本中，我们的攻击成功率超过90%。

有关此实验的更多详细信息，请参见附录D。

5 前置投毒

我们的第二次攻击消除了对敌方在训练集中对网络数据具有持续控制的假设。为了做到这一点

我们做出了一个新的假设：我们可以准确预测网页内容何时被下载。我们将在基于维基百科的数据集上进行研究，但也会讨论附录B中可能存在类似漏洞的Common Crawl数据集。

5.1 我们的攻击：编辑维基百科

维基百科是一个众包百科全书。这使得它成为互联网上最全面和可靠的数据集之一[79]。由于其质量和多样性，维基百科经常被用作机器学习训练数据的来源。事实上，许多语言建模数据集严重依赖于英文维基百科，例如，它在BERT训练集中占据了75%以上的词汇[21]，在Pile数据集中占据了1.5%[23]，以及在WikiText数据集中占据了全部内容[43]。许多任务特定的数据集也依赖于英文维基百科，例如，WikiQA [81]问题回答数据集（30,000+次下载）和WikiBio [38]传记写作数据集（19,000+次下载）。最后，一些在第4节讨论的分布式数据集中索引了许多维基百科文章中的图片。

由于维基百科是一个任何人都可以编辑的实时资源，攻击者可以通过进行恶意编辑来污染从维基百科获取的训练集。蓄意的恶意编辑（或“破坏行为”）在维基百科上并不罕见，但通常会在几分钟内手动恢复[80]。因此，实际上污染维基百科是具有挑战性的：与我们之前的攻击不同，攻击者无法持续控制任何特定页面，因此他们必须希望他们的恶意编辑在被恢复之前恰好影响到数据集的下载。

然而，我们做出了一个关键观察，这将确保我们的投毒攻击成功：从维基百科衍生的数据集本身并不是实时的，而是一系列静态快照。这是因为维基百科禁止使用网络爬虫来抓取实时网站。相反，维基百科提供整个百科全书的定期“转储”（或快照）。因此，从维基百科获取的训练数据集使用这些快照，而不是直接从网站抓取的数据。

例如，BERT模型的作者[21]明确建议“下载最新的[Wikipedia]转储”以重现他们的结果。

这使得我们可以进行所谓的前置攻击。一个攻击者如果能够预测何时Wikipedia页面将被抓取并包含在下一个快照中，就可以在抓取之前立即进行投毒。即使编辑在实时页面上迅速恢复，快照仍然会包含恶意内容—永远。细心的读者可能会认为我们没有获得太多：攻击者不再需要预测终端用户抓取Wikipedia的时间来生成训练集，而是需要预测Wikipedia被抓取以生成官方快照的时间。

但正如我们将看到的，后者实际上很容易。

在本节中，我们将探讨对手如何计时恶意操作。

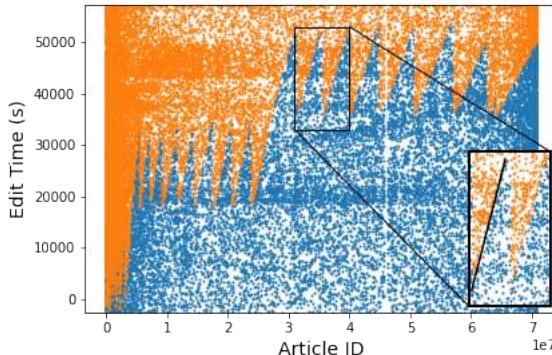


图3：对手可以轻松预测任何给定的Wikipedia文章将在两个月一次的转储中被快照。我们可视化了2022年6月1日Wikipedia快照周围的编辑。每个点对应于对Wikipedia文章的编辑，X轴上是文章ID，Y轴上是编辑的时间（以秒为单位）。蓝色的编辑点被包含在快照中，橙色的编辑点则未被包含。图中展示的“锯齿”模式表明存在一种趋势，即多个并行作业按顺序抓取Wikipedia文章以构建快照。此外，这些并行作业几乎完美地线性地运行其分配的页面。

为了确保成功地对维基百科快照进行投毒，需要进行恶意编辑。为此，我们需要回答两个问题：

1. 我们能够准确预测维基百科快照中页面被抓取的时间吗？
2. 恶意编辑会被多快地撤销？

5.2 预测检查点时间

维基百科使用一种确定性的、有详细文档的协议来生成快照（通过检查可以轻松逆向工程）。这使得可以高精度地预测单个文章的快照时间。

5.2.1 维基百科快照的工作原理

英文维基百科每个月的1日和20日进行归档。快照是由 n 个并行工作者生成的；所有维基百科文章按照ID顺序排列，并分成 n 个块，每个工作者独立地线性抓取其块中的所有文章。

由于维基百科的规模，整个过程需要近一天的时间才能完成。因此，不同的文章在墙上时钟时间上被抓取的时间有很大差异。因此，一个文章在时间 $t_{i,j}$ 进行的编辑可能会被排除在快照之外，而另一篇文章在时间 $t_{i,j} > t_{i,j}$ 进行的编辑可能会被包括在内。图3展示了这种“锯齿”效应：在6月1日的快照中有很多编辑（蓝色）被包括进去。

在不包括（橙色）的不同编辑之前，进行了以下编辑

要使一个前置攻击成功，仅仅预测快照过程开始的时间是不够的。
攻击者还需要预测每个单独页面被抓取的精确时间。

5.2.2 利用滚动快照

为了精确预测每篇文章的抓取时间，攻击者可以利用维基百科快照过程中的一致性。

首先，对手准确地知道每个转储开始的时间，因为维基媒体通过发布正在进行的快照的实时统计信息来提供这些信息。其次，文章在一个转储中被爬取的速率

在各个转储之间保持几乎一致，并且因此可以从之前的转储中近似估计（有趣的是，爬行速度随时间略微加快）。

有了这两个信息，攻击者可以精确预测任何给定文章的爬取时间。对于一篇文章 i ，我们将其在当前快照中被爬取的时间表示为 t_i ，并将其在上一个快照中的爬取时间表示为 $t_{i,\text{prev}}$ 。我们将当前和上一个快照的开始时间（由维基媒体报告）分别表示为 t_0 和 $t_{0,\text{prev}}$ 。根据我们上面的第一个观察，攻击者知道 t_0 和 $t_{0,\text{prev}}$ 。根据我们的第二个观察，我们有 $t_i - t_0 \approx t_{i,\text{prev}} - t_{0,\text{prev}}$ 。这使我们能够估计第 i 篇文章的快照时间为 $t_i \approx t_0 + (t_{i,\text{prev}} - t_{0,\text{prev}})$ 。

但是计算这个需要知道 $t_{i,\text{prev}}$ ——上一个快照中抓取第 i 篇文章的时间。
现在我们讨论如何追溯估计这个时间。

5.2.3 确定文章快照时间

维基百科快照并没有明确列出每篇文章的快照时间。但是维基百科提供了一些辅助信息：一个包含每次编辑的精确时间的完整编辑列表。我们展示如何利用这些信息来追溯估计一篇文章的快照时间。

回顾图3，对于每篇文章，我们可以找到包含在当前快照中的编辑列表（蓝色点），后续编辑出现在下一个快照中（橙色点）。因此，对于每篇文章，我们知道快照时间 t_i 在文章最后一个包含的编辑（最上面的蓝色点）和第一个不包含的编辑（最下面的橙色点）之间。然而，这个时间间隔是宽松的：这些编辑之间的时间通常相隔几天。

为了改进我们对每篇文章快照时间的估计，我们可以再次利用维基百科爬取过程中的一致性。我们观察到文章是按顺序处理的：通过放大到仅一个爬取作业，如图3所示，我们可以看到文章是按顺序爬取的，并且有一条明确的线将最后包含的文章和第一个

⁸在<https://dumps.wikimedia.org/backup-index.html>

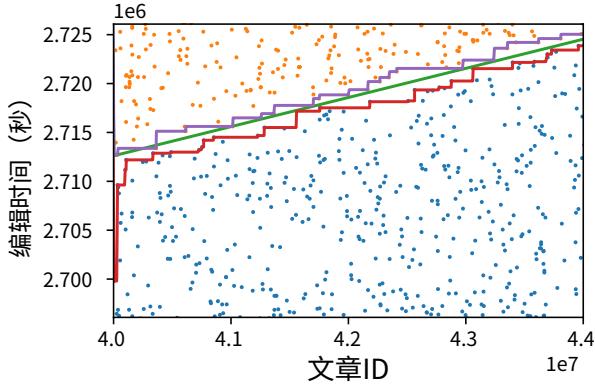


图4：我们可以对每篇文章的快照时间获得紧密的估计。绿色和橙色线表示间隔 $[t_{i,prev}^{low}, t_{i,prev}^{high}]$ 对于英文维基百科的一系列文章。平均而言，我们的预测（蓝线）距离最远的间隔边界有27分钟的距离。

每篇文章的未包含编辑。也就是说，对于在同一个作业中处理的文章 i 和 j ，如果 $i < j$ ，则有 $t_i < t_j$ 。因此，我们可以通过不断跟踪快照之前最近的编辑（对于每篇文章，这是该作业中较早的文章上进行的最高蓝色编辑），以及作业中所有后续文章中未包含在快照中的最早编辑，来收紧每篇文章的编辑时间间隔。我们在图4中可视化了这一点。对于上一个快照中的每篇文章，我们可以获得一个时间间隔。

$[t_{i,前}^{低}, t_{i,前}^{高}]$ 包含真实（但未知）快照时间 $t_{i,前}$ 。根据我们上面概述的构造，我们保证这些区间的下限和上限在作业中的所有文章中单调递增（见图4）。

为了为每篇文章的先前快照时间产生一个估计 $\hat{t}_{i,前}$ ，我们计算单个线程处理的所有文章的快照区间的最佳线性拟合，如图4所示。这样我们可以预测文章的下一个快照时间为 $\hat{t}_i \approx t_0 + (\hat{t}_{i,前} - t_0, 前)$ 。

5.2.4 评估我们的预测

现在我们评估我们估计文章快照时间的过程。理想情况下，我们将直接将我们预测的快照时间 \hat{t}_i 与第 i 篇文章的真实快照时间进行比较。但是，除了某个区间之外，我们不知道真实情况。

$[t_{i,前}^{低}, t_{i,前}^{高}]$ 我们可以根据上述描述来计算后验。因此，我们分两步进行。

首先，我们展示了我们的线性拟合来估计先前的快照时间 $\hat{t}_{i,prev}$ 是准确的。为此，我们测量了预测时间 $\hat{t}_{i,prev}$ 和未知的真实值之间的最大绝对误差，在区间 $[t_{i,prev}^{low}, t_{i,prev}^{high}]$ 内。这提供了真实估计误差的上界。我们发现，估计误差平均上限为27分钟。

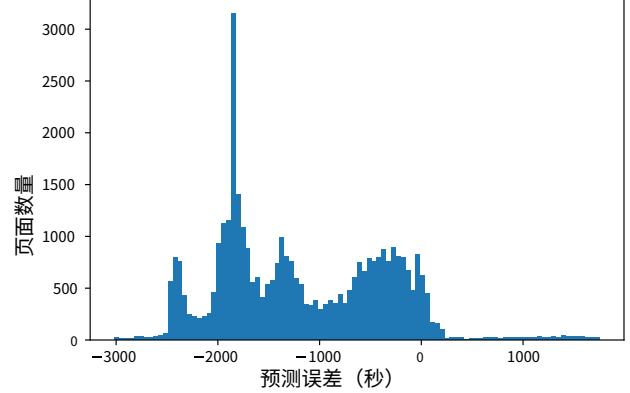


图5：维基百科检查点时间预测误差的分布。大多数预测的检查点时间与我们构建的真实值相差不超过30分钟。总体上，我们过早预测了编辑时间，因此后续需要进行调整，我们将在第5.4节中讨论。

其次，我们评估了从一个快照到下一个快照的外推的准确性。也就是说，我们评估了我们的先验预测的快照时间 $\hat{t}_i := t_0 + (\hat{t}_{i,prev} - t_0, prev)$ 与我们可以事后估计的快照时间之间的接近程度，后者使用了上述线性拟合描述。

图5显示了误差估计的分布。在大多数情况下，我们的预测准确度在大约30分钟以内。然而，我们注意到，我们的外推误差有一定的负向偏差。我们发现这是因为快照随时间略微加速，所以我们通常会高估一篇文章的下一个快照时间。在第5.4节中，当我们对维基百科快照的投毒攻击的成功进行保守估计时，我们将纠正这一点。

5.3 估计修订速度

现在我们已经测量了我们能够准确预测未来快照发生的时间，我们将注意力转向测量在恶意编辑被还原之前的机会窗口的大小。

我们注意到，虽然最准确的方法是注入恶意编辑并测量还原时间的分布，但我们认为这是不道德的。相反，我们采取了一种完全被动的——尽管不太准确的——方法，如第5.6节所讨论的。

为了测量修订的速度，我们构建了一个数据集，其中包含从2021年1月到2022年6月（共18个月）对维基百科进行的所有编辑，并将每个编辑分类为添加或还原，如果它们包含一组固定的字符串⁹，这些字符串经常用于还原评论。然后，我们保守地假设正在编辑的内容

⁹ 这组字符串是通过对维基百科每个评论样本进行手动分析生成的；详细信息请参见附录B.1。

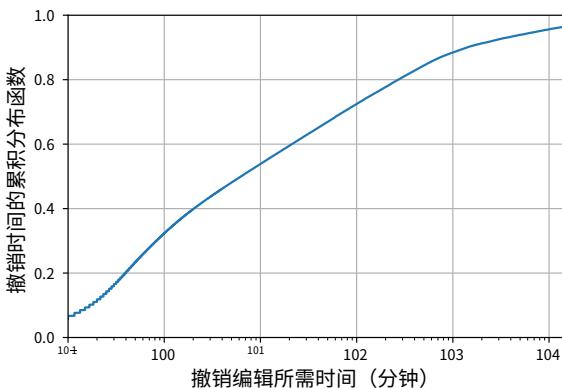


图6：英文维基百科修订时间的累积分布函数。
大约35%的修订时间超过30分钟。

撤销的是紧接在前的编辑，因此我们将撤销时间定义为这两个编辑之间的时间间隔。¹⁰图6绘制了这个分布。当我们预测未来快照时间的大约30分钟误差（参见图5）与我们对真实快照时间估计的平均不确定性的另外大约30分钟（参见图4）结合起来时，我们可以保守地估计攻击者可以根据真实快照时间的平均值在一小时内进行编辑。大约32%的撤销时间超过一个小时，因此攻击很可能经常成功。在下一节中，我们将进一步细化这个估计，以更准确地确定我们可以投毒的文章数量。

5.4 将所有内容整合起来

利用我们对相对文章快照时间的预测，对真实快照时间的区间界限以及回退时间的分布，我们现在可以（保守地）确定对维基百科进行投毒的对手可能占据的比例。

有两种潜在的“失败情况”，恶意编辑可能无法进入检查点：

- 恶意编辑应用得太晚：文章已经快照，或者
- 恶意编辑应用得太早：编辑在文章快照之前被撤销

这引发了一个权衡：攻击者希望尽早进行编辑，以确保不错过快照时间，但又希望晚一些以最大化抢先编辑者的机会。

因此，我们按照以下方式计算应用恶意编辑的最佳时间。请记住，我们使用 $[t_i^{\text{low}}, t_i^{\text{high}}]$ 来表示真实（但未知）快照时间周围最紧密的区间。

¹⁰这会低估编辑时间，因为如果破坏行为是在之前的编辑中进行的，我们会错误地使用后来编辑的时间。

t_i 的文章，第 i 篇文章的预测快照时间为 \hat{t}_i 。
为了平衡上述两种故障模式，并考虑我们预测中的偏差（参见第5.2.4节），我们引入了一个“调整”变量 a ，使得对手在时间 $\hat{t}_i + a$ 上添加恶意编辑，而不是恰好在时间 \hat{t}_i 上。

那么，当恶意维基百科编辑在时间 $\hat{t}_i + a$ 进行时，进入快照的恶意编辑的比例可以作为下界：

$$\mathcal{A}(a) = \frac{1}{|D|} \sum_{i \in D} \underbrace{(1 - p_{\text{rev}}(\hat{t}_i + a; t_i^{\text{high}}))}_{\substack{\text{编辑应用过早} \\ \text{编辑应用太晚}}} \cdot (1 - \underbrace{1[\hat{t}_i + a > t_i^{\text{low}}]}_{\substack{\text{编辑应用太早} \\ \text{编辑应用过晚}}}) ,$$

其中 $1[\hat{t}_i + a > t_i^{\text{low}}]$ 是指示函数，如果编辑在检查点之后应用（在这里我们保守地使用真实检查点时间的下界 t_i^{low} ），则为1

$p_{\text{rev}}(\hat{t}_i + a; t_i^{\text{high}})$ 是编辑在检查点之前被还原的概率（在这里我们保守地使用上界 t_i^{high} ）

关于真实检查点时间，我们使用第5.2节和第5.3节的结果计算这个和。通过在潜在的 a 值的扫描中取最大值，我们得到 $\max_a \mathcal{A}(a) = 0.065$ 。也就是说，根据我们的保守分析，我们可以在没有其他防御措施的情况下，污染6.5%的维基百科文档。

当然，在现实中，除了我们的分析之外，还有许多因素可能会阻止我们达到这个比例，比如编辑速率限制或IP封禁。我们在选择调整值 a 的最优值时也会“作弊”，但我们不认为这是一个主要限制因素——很可能对手可以使用更多的历史数据来产生更好的估计 \hat{t}_i 以及对 a 的良好估计。然而，我们的分析也是悲观的，因为我们假设我们只尝试一次来污染任何给定的文章。对手可能会尝试更有针对性的攻击，正如我们在附录B.2中讨论的那样，他们会对目标文章进行多次编辑尝试，迫使编辑者多次回滚，并增加编辑成功的可能性。最终，我们对投毒成功率的最大努力估计比以前的投毒攻击[15]所需的数量级高得多。因此，我们认为在维基百科快照上成功进行前置投毒攻击是可行的，并且寻找减轻此类攻击的方法是一个值得研究的方向。

5.5 多语种维基百科

维基百科也经常用于非英语语言建模。例如，BERT的多语种版本完全是在前104个维基百科语言上训练的。¹¹多语种数据集通常比英语数据集更依赖于维基百科。因此，对非英语语言建模任务来说，污染维基百科甚至更加有害。为了衡量

¹¹请参阅 <https://github.com/google-research/bert/blob/master/multilingual.md#list-of-languages>。

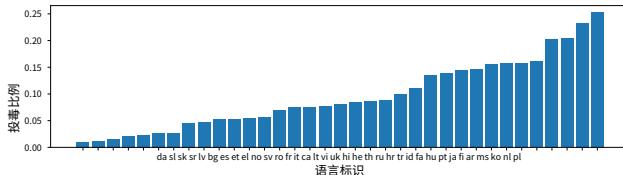


图7：多语种维基百科可能更容易受到前置攻击的影响。我们通过重复使用第5.2到5.4节中的攻击方法，计算了包含在Wiki-40B [25]中的36种语言的投毒率。

在这个漏洞中，我们调查了经常用于训练大型多语言模型的Wiki-40B数据集[25]。

我们在Wiki-40B中的39种非英语语言中重复了上一节的分析，通过识别哪些字符串经常表示这些语言中的还原。¹²再次强调，我们的分析是松散的：我们只识别了一部分（通常是自动化的）还原；然而出于与上述相同的原因，我们相信这代表了平均还原时间的下限。

我们发现22个（63%）非英语维基百科比英语维基百科更容易被投毒，如图7所示。可行的投毒率范围从0.95%到25.3%，中位数为8.2%。总体上，脆弱性的增加来自于多语言维基百科具有更可预测的检查点，原因有两个。首先，由于这些维基百科较小，整个检查点过程较短，减少了不同页面之间检查点时间的差异。其次，由于维基百科在连续检查点之间的变化较小，检查点的速度更加稳定，提高了我们的预测能力。这可能是为什么一些较大的维基百科，如西班牙语、丹麦语和意大利语，与英语维基百科具有可比的投毒率。然而，对于编辑速度较慢的语言，我们的基于区间的测量也更加保守，因为区间会更大，给出了一些小维基百科的非常小的下限，比如斯洛伐克语和斯洛文尼亚语。

我们再次强调，由于IP封禁或速率限制，这种大规模投毒的可能性是不太可能发生的。我们在分析中得出的最重要结论是：1) 多语种维基百科容易受到投毒攻击，而且通常比英语维基百科更容易受到攻击；2) 多语种数据集更倾向于依赖维基百科而不是英语数据集，从而增加了这种风险。

5.6 伦理考虑

我们在这里的行动完全是被动的。我们对维基百科没有进行任何编辑，除了从官方来源下载数据集之外-

¹²我们无法在我们研究的检查点时间内访问德语、中文和捷克的检查点，并且塔加洛语没有足够的数据进行可靠的分析。

我们从未与维基百科互动，只从官方来源下载数据集。虽然这在一定程度上限制了我们的分析，但我们认为这是进行此类研究的正确方式，以避免对维基百科编辑社区造成伤害。我们向维基媒体的研究人员披露了我们的攻击分析（以及后来的防御建议），他们在本文发布之前承认了这个漏洞。

6 防御措施

为了解决我们发现的攻击，我们提出了一种基于完整性的分割视图投毒防御和一种基于时间的前置投毒防御。我们还讨论了解决投毒问题的潜在方向。我们将这些防御措施与数据集的管理者、维护者和下载者共享，作为我们负责任的披露的一部分，并报告了他们对防御措施的实施情况。

6.1 现有的信任假设

根据我们的威胁模型（第3节），我们提出的防御措施假设所有的维护者、管理者和下载者都是可信的，并且会诚实行为。这意味着维护者向任何客户端提供相同的分布式数据集索引 $\{(url_i, c_i)\}_{i=1}^{Ni}$ ，并且索引本身没有被投毒（例如，由于内部风险或被攻击）。下载者对于分布式数据集会诚实地访问所有的 url_i ，并通过我们的防御措施计算任何完整性检查。管理者向所有客户端提供相同的集中式数据集 \mathcal{D} ，管理者控制任何元素 url_i 的快照时间 t_{io} 。

这些假设反映了客户对维护者、策展人和下载者的现有信任，并因此代表了实施防御的最快、短期的途径。我们在第6.5节中讨论了这些信任假设的局限性和更少信任假设的更强大解决方案。

6.2 防止分割视图投毒

防止分割视图投毒攻击的最简单的方法是将分布式数据集 $\{(网址_i, 内容_i)\}_{i=1}^{Ni}$ 转换为集中式数据集（例如，像YFCC100M [71]中那样），但由于在第3节中提到的经济、隐私和法律挑战，目前这是不现实的。相反，维护者或其他可信第三方可以通过在任何攻击之前附加一个来自网址 i 的原始数据 x_i 的加密哈希 $h_i = H(x_i)$ 来防止分割视图攻击。然后下载者将检查是否 $H(x'_i) = h_i$ ，其中 x'_i 是网址 i 在时间 t_i 时的内容。下载者会丢弃客户和维护者接收到不同内容的任何数据。在这里， H 应该是一个像SHA-256这样的加密哈希函数。

实施和负责任的披露。实施这种防御需要进行一系列的生态系统变化。目前，只有PubFig和FaceScrub在其分布式数据集中包含了加密哈希（见表1）。而且，由于

这两个数据集没有提供官方的下载器，社区一直依赖于许多第三方下载脚本，这些脚本大部分实际上并没有验证这些哈希值。¹³幸运的是，对于较大的数据集，img2dataset [5]工具已成为75%的请求中使用的规范下载器。

作为我们负责任的披露的一部分，我们联系了每个维护者（见表1），建议将SHA-256哈希作为数据集索引的一部分。在撰写本文时，CC3M、CC12M、LAION-2B-en、LAION-2b-multi、LAION-1B-nolang和LAION-400M现在发布其带有图像内容的SHA-256哈希的数据集。

我们还在img2dataset中实现了一个选项，可以在下载时验证SHA-256图像哈希，从而防止我们的攻击，为使用此工具的任何人提供我们自己的哈希备份，存储在Google Cloud Bucket的gs://research/distributed-dataset-hashes中，用于我们拥有（近乎）原始数据的数据集。

限制。如果大部分良性内容保持不变，完整性检查是可行的。如果内容以任何方式被更改（例如重新编码、裁剪、调整大小或上传更高分辨率的图像），原始哈希将不再匹配。这可能会严重降低数据集的效用：例如，我们从2018年下载的第一个Conceptual Captions 3M数据集中获取原始原始数据，并将其与我们在2023年最新下载的相同图像进行比较。

在330万张原始图像中，仍有290万张图像托管在线，但只有110万张图像的哈希与原始哈希匹配，其他180万张图像自初始数据集构建以来发生了变化。

这表明，尽管我们的防御措施可以完美地防止分割视图投毒攻击，但可能会显著降低效用。在附录C中，我们对PubFig和FaceScrub数据集进行了案例研究分析，结果显示，虽然存在无效哈希的图像占比较大，但仍然包含了修改后但有用的内容。转换为感知哈希函数（旨在对小图像变化具有不变性）会提高效用，但并不能有效地防止我们的投毒攻击，因为攻击者可以上传经过对抗性修改的毒害图像来欺骗感知哈希[26,29,68]。这表明，为了在不牺牲效用的情况下防御我们的攻击，需要提出全新的防御思路。

6.3 防止前置投毒

我们的前置投毒攻击依赖于一个事实，即攻击者只需要在几分钟内持续控制数据即可成功。为了防御这种攻击，增加

¹³我们检查了每个数据集的6个最流行的下载脚本（通过搜索“[pubfig|facescrub] 数据集下载github”获得），发现每个数据集只有一个脚本实现了哈希验证。FaceScrub的脚本默认检查哈希值，而PubFig的脚本要求用户运行一个单独的验证脚本。

攻击者必须在时间 t_i 内保持对 url_i 的控制，才能将其包含在快照中，其中 t_i 表示攻击者首次修改URL内容的时间。

如果策展人能够在时间 Δ 内检测到恶意修改，那么增加 $d > \Delta$ 可以有效阻止攻击。这可以通过两种方式实现：

(1) 策展人可以随机化 url_i 的快照顺序，并延长所需的快照时间；或者 (2) 策展人可以在时间 t_i 冻结 url_i 的内容编辑，等待一个时间段 $T > \Delta$ 以便编辑通过审核，然后最终在时间 $t_i + T$ 发布快照。

实施和负责任的披露。对于我们的第一个方法，维基百科可以随机化其文章的快照顺序，而不是基于文章ID的当前顺序方法。这样可以阻止对手准确预测文章何时被选择进行快照，需要他们在整个快照时间内控制文章，以确保成功。对于英文维基百科，检测破坏行为的当前平均审查时间为2.5小时（图6）。将快照时间延长超过 Δ 将保护 $1 - \Delta / (t_n - t_0)$ 的文章免受随机的恶意修改，或者如果快照在24小时内均匀随机化，则保护89.5%的文章。这假设攻击者无法使用Sybil账户在第一次检测和还原后自动重新引入恶意编辑。如果这个假设无效，实际上这种保护将会更加脆弱。

对于我们的第二种更全面的方法，维基百科可以创建一篇文章的初始快照，保留一段时间 $T > \Delta$ ，然后从可信的管理员那里回溯（“挑选”）修改或还原在最终确定快照之前发生在时间 T 之前的修改。（后续编辑必须由可信的管理员接受，以避免攻击者的选择性删除或还原。）即使是一个合理的宽限期一天也可能对捕捉到的恶意编辑数量产生重大影响。

例如，在英文维基百科上（图6），将窗口从5分钟增加到1天，将会将还原率从50%增加到90%，将破坏行为减少了5倍。

作为我们负责任的披露的一部分，我们通知了维基百科这些攻击和我们提出的防御措施。

限制。在实践中，这些防御措施使攻击者更难以操作前沿，但无法完全阻止，因为 Δ 在文章之间不均匀。例如，攻击者可能会针对活动较少的文章或拥有较少的语言管理员的文章，以提高他们的前沿成功率。此外，我们的防御措施依赖于存在一个可信的策展人，可以检测恶意编辑，这可能是困难的，如果攻击者有意引入难以察觉的变化，只影响机器理解，但对人工审查来说是有效的。克服这些风险——对于任何“活跃”数据集都存在的风险——需要更复杂的解决方案，我们将在下面探讨。

6.4 防止一般的投毒

防止对更一般的网规模数据集（如Common Crawl，一个拥有peta字节大小的网络爬行数据集）的投毒攻击更加复杂。这里没有可信的“黄金”快照，就像我们对分割视图投毒时那样。也没有一个可信的策展人可以检测恶意编辑。同样令人担忧的是，网页的更新版本之间没有现实的界限，可以作为信号来附加信任。最终，对于哪些域名值得信任的任何概念都是临时的。

因此，客户可以依赖基于共识的方法（例如，仅在许多不同的网站上出现的图像标题对）。然后，攻击者必须污染足够多的类似网站，以确保成功，这与分布式系统（如区块链）中存在的共识挑战相似[45]。然而，这个领域的任何解决方案都需要对URL内容在训练过程中如何被消费、向量化和解决冲突有下游知识。我们将应用特定的解决方案留给未来的工作，以解决一般的投毒问题。

6.5 透明度以提高信任

如今，网规模数据集依赖于隐式的信任。客户信任维护者分发相同和准确的辅助数据 c_i ，但实际上可能是恶意的，因为维护者受到了威胁。客户信任策展人能够有效地进行审查，以检测对 x_i 的恶意编辑。客户信任下载者能够准确地检索 url_i 。最后，客户信任网站为每个 url_i 提供相同的 x_i ，尽管攻击者有无数机制来破坏 x_i ，不仅仅是购买过期的域名或抢先快照。

我们相信改善网规模数据集的安全性需要在生态系统中引入透明度。数据透明度围绕着分布给客户端的集合 $\{(url_i, c_i, h_i)\}$ ，类似于证书透明度[37]，可以防止短暂故障或被入侵的维护者将不同的数据集分发给不同的客户端，并有助于检测和删除随时间变化的不准确 c_i 或过期 url_i 。策展人可以进行类似的过程，以确保所有客户端都收到相同的语料库 \mathcal{D} 。虽然许多下载器已经是开源的，但二进制透明度将增强保护，以防止恶意模块的选择性包含[1]。这种透明度将为生态系统做好准备，以应对多个维护者和策展人不断更新网规模数据集的未来，而不是目前对集中实体和静态数据集的依赖。

7 结论

我们的论文证明了网规模数据集容易受到低成本和极其实用的投毒攻击，这些攻击甚至可以在今天实施。即使攻击者

可以只针对一小部分经过筛选的数据集进行攻击，在这些数据集中，污染0.01%的样本就足以对模型进行投毒。那些发布和维护数据集的人应该考虑我们介绍的防御措施，包括完整性检查和随机或时间限制的快照，或者选择其他应用特定的防御措施。根据我们的发现，我们认为机器学习研究人员必须重新评估他们对网规模数据的信任假设，并开始探索不依赖于单一信任根源的解决方案。我们的发现还揭示了攻击研究的各种未来方向：威胁模型中，攻击者只能编辑原始内容而不能编辑辅助数据，如标签；评估提出的攻击的实际成本；评估更宽松但潜在易受攻击的近似完整性检查的有效性。因此，我们的工作只是社区开发对从网规模数据生成模型所涉及的风险有更好理解的起点。

致谢

我们感谢数据集策展人（特别是BeerChangpinyo、Saehoon Kim、Romain Beaumont、Ludwig Schmidt和Chris Albon）对数据集和防御措施的讨论。我们还要感谢Milda Nasr和AlexKurakin对本文初稿的评论。

参考文献

- [1] Mustafa Al-Bassam和Sarah Meiklejohn。Contour: 一个实用的二进制透明系统。在数据隐私管理、加密货币和区块链技术，第94-110页。Springer，2018年。
- [2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak和Zachary Ives。DBpedia: 一个开放数据网络的核心。在语义Web，第722-735页。Springer，2007年。
- [3] Ankan Bansal, Carlos Castillo, Rajeev Ranjan和Rama Chellappa。基于CNN的人脸验证的要与不要。在IEEE国际计算机视觉会议工作坊的论文集中，第2545-2554页，2017年。
- [4] Ankan Bansal, Anirudh Nanduri, Carlos D Castillo, Rajeev Ranjan, 和 Rama Chellappa。UMDfaces：用于训练深度网络的注释人脸数据集。在2017年IEEE国际联合会议上生物识别（IJCBA），页码464-473。IEEE，2017年。
- [5] Romain Beaumont。img2dataset：轻松将大量图像URL转换为图像数据集。<https://github.com/rom1504/img2dataset>, 2021年。
- [6] Battista Biggio, Blaine Nelson, 和 Pavel Laskov. 对支持向量机的投毒攻击。arXiv预印本arXiv:1206.6389, 2012年。
- [7] Abeba Birhane, Vinay Uday Prabhu, 和 Emmanuel Kahembwe. 多模态数据集：厌恶、色情和恶性刻板印象。arXiv预印本arXiv:2110.01963, 2021年。
- [8] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, 和 Adam T Kalai。男人是计算机程序员，女人是家庭主妇吗？去偏置词嵌入。神经信息处理系统的进展，29，2016年。
- [9] Antoine Bordes, Nicolas Usunier, Sumit Chopra 和 Jason Weston。大规模简单问题回答与记忆网络。arXiv预印本arXiv:1506.02075, 2015年。

- [10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell 等。语言模型是少样本学习器。神经信息处理系统的进展, 33:1877–1901, 2020年。
- [11] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woon-hyuk Baek, and Saehoon Kim. COYO-700M: 图像-文本对数据集。<https://github.com/kakaobrain/coyo-dataset>, 2022年。
- [12] Qingqing Cai and Alexander Yates. 通过模式匹配和词典扩展进行大规模语义解析。在计算语言学协会第51届年会论文集(第1卷 : 长篇论文), 页码423–433, 2013年。
- [13] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. VGGFace2: 一个用于识别不同姿势和年龄的数据集。在2018年第13届IEEE国际自动人脸与手势识别会议(FG 2018), 页码67–74。IEEE, 2018年。
- [14] Nicholas Carlini. 毒害半监督学习的未标记数据集。在第30届USENIX安全研讨会 (USENIX Security 21) 中, 页码1577–1592, 2021年。
- [15] Nicholas Carlini和Andreas Terzis. 毒害和后门 对比学习。arXiv预印本arXiv:2106.09667, 2021年。
- [16] Soravit Changpinyo, Piyush Sharma, Nan Ding和Radu Soricut. 概念12M: 推动网规模图像文本预训练以识别长尾视觉概念。在IREE/CVF计算机视觉和模式识别会议论文集中, 页码3558–3568, 2021年。
- [17] Honglie Chen, Weidi Xie, Andrea Vedaldi和Andrew Zisserman. VGGsound: 一个大规模的音频-视觉数据集。在ICASSP 2020-2020 IEEE国际声学、语音和信号处理会议 (ICASSP) 中, 页码721–725。IEEE, 2020年。
- [18] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 使用数据投毒对深度学习系统进行有针对性的后门攻击。arXiv预印本arXiv:1712.05526, 2017年。
- [19] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann等。PaLM: 使用路径扩展语言建模的规模化方法。arXiv预印本arXiv:2204.02311, 2022年。
- [20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: 一个大规模的分层图像数据库。在2009年IEEE计算机视觉和模式识别会议, 第248–255页。Ieee, 2009年。
- [21] Jacob Devlin, Ming-Wei Chang, Kenton Lee和Kristina Toutanova。BERT: 用于语言理解的深度双向变电器的预训练。在2019年北美计算语言学协会会议论文集: 人类语言技术, 第1卷 (长篇和短篇), 4171–4186页, 明尼阿波利斯, 明尼苏达州, 2019年。计算语言学协会。
- [22] Mohin Dubey, Debayan Banerjee, Abdelrahman Abdelkawi和Jens Lehmann。LC-QuAD 2.0: 用于Wikidata和DBpedia的复杂问题回答的大型数据集。在国际语义网会议, 第69–78页。Springer, 2019年。
- [23] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, 等。The Pile: 用于语言建模的800GB多样化文本数据集。arXiv预印本arXiv:2101.00027, 2020年。
- [24] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: 评估深度神经网络的后门攻击。IEEETweet Access, 7:47230–47244, 2019年。
- [25] Mandy Guo, Zihang Dai, Denny Vrandečić, and Rami Al-Rfou. Wiki-40B: 多语言语言模型数据集。在第12届语言资源和评估会议论文集, 页码2440–2452, 2020年。
- [26] Qingying Hao, Licheng Luo, Steve TK Jan, and Gang Wang. 它并不像看起来那样: 操纵基于感知哈希的应用程序。在2021年ACM SIGSAC计算机和通信安全会议论文集, 2021年。
- [27] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchats kaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne He ndricks, Johannes Welbl, Aidan Clark, 等。训练计算优化的大型语言模型。arXiv预印本arXiv:2203.15556, 2022年。
- [28] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, 和Ludwig Schmidt. OpenCLIP, 2021年7月。
- [29] Shubham Jain, Ana-Maria Cretu, 和Yves-Alexandre de Montjoye. 对抗性检测规避攻击: 评估基于感知哈希的客户端扫描的鲁棒性。在第31届USENIX安全研讨会 (USENIX Security 22) , 第231 7–2334页, 2022年。
- [30] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 神经语言模型的规模定律。arXiv预印本arXiv:2001.08361, 2020。
- [31] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Eva nBrossard. MegaFace基准: 100万张人脸的规模识别。在计算机视觉和模式识别IEEE会议论文集, 2016年, 第4873–4882页。
- [32] Evan Klinger and David Starkweather. pHASH: 开源的感知哈希库。<https://phash.org/>, 2013年。
- [33] Alex Krizhevsky. 从微小图像中学习多层特征, 2009年。
- [34] Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. 用于人脸验证的属性和类比分类器。在2009年IEEE第12届国际计算机视觉会议上, 第365–372页。IEEE, 2009年。
- [35] Tobias Lauinger, Ahmet S Buyukkayhan, Abdelberi Chaabane, Williamson Robertson, and Engin Kirda. 从删除到重新注册, 零秒钟: 域名注册商的行为。在互联网测量会议的论文集中, 2018年。
- [36] Tobias Lauinger, Abdelberi Chaabane, Ahmet Salih Buyukkayhan, Kaan Onarlioglu, and William Robertson. 注册商之战: 过期后的域名接管的实证分析。在第26届USENIX安全研讨会 (USENIX Security 17) 上, 第865–880页, 温哥华, BC, 2017年8月。USENIX协会。[37] Ben Laurie. 证书透明。ACM通信, 57 (10) : 40–46, 2014年。
- [38] Rémi Lebret, David Grangier和Michael Auli. 生成文本 从结构化数据到传记领域的应用。CoRR, abs/1603.07771, 2016年。
- [39] Chaz Lever, Robert Walls, Yacin Nadji, David Dagon, Patrick McDaniel和Manos Antonakakis。Domain-Z: 28个注册后测量在域中利用剩余信任。在2016年IEEE安全与隐私研讨会 (SP) , 页691–706。IEEE, 2016年。
- [40] Alexandra Lucchini和Joseph Viviano. 盒子里有什么? 对Common Crawl语料库中的不良内容进行分析。在计算语言学协会第59届年会和第11届国际联合会议自然语言处理 (第2卷: 短论文) , 页782–789, 2021年。
- [41] Mary Ann Marcinkiewicz. 构建一个大型的英语注释语料库: 宾州树库。使用大型语料库, 273, 1994年。
- [42] Pablo Mendes, Max Jakob, 和Christian Bizer. DBpedia: 一个多语言跨领域的知识库。在第八届国际语言资源与评估会议 (LREC'12) , 页码1813–1817, 土耳其伊斯坦布尔, 2012年5月。欧洲语言资源协会 (ELRA) 。

- [43] Stephen Merity, Caiming Xiong, James Bradbury, 和 Richard Socher. 指针哨兵混合模型, 2016年。
- [44] Tyler Moore 和 Richard Clayton. 银行过去的幽灵: 对关闭的银行网站的实证分析。在国际会议金融密码学与数据安全, 页码33-48。Springer, 2014年。
- [45] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, 和 Steven Goldfeder. 比特币和加密货币技术: 全面介绍。普林斯顿大学出版社, 2016年。
- [46] 吴宏伟和斯特凡·温克勒。一种基于数据驱动的方法来清理大型人脸数据集。在2014年IEEE国际图像处理会议 (ICIP), 页码343-347。IEEE, 2014年。
- [47] 尼克·尼基福拉基斯, 卢卡·因弗尼齐, 亚历山德罗斯·卡普拉维洛斯, 史蒂芬·阿克尔, 沃特·乔森, 克里斯托弗·克鲁格尔, 弗兰克·皮森斯和乔瓦尼·维尼亚。你是你所包含的内容: 对远程JavaScript包含的大规模评估。在2012年ACM计算机和通信安全会议论文集, 页码736-747, 2012年。
- [48] OpenAI。介绍Whisper。 <https://openai.com/blog/whisper/>, 2022年。
- [49] Omkar M Parkhi, Andrea Vedaldi和Andrew Zisserman。深度人脸识别。2015年。
- [50] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, 等。从自然语言监督中学习可迁移的视觉模型。在机器学习国际会议上, 页码为8748-8763。PMLR, 2021年。
- [51] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey和Ilya Sutskever。通过大规模弱监督实现鲁棒语音识别。
- [52] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever等。语言模型是无监督的多任务学习者。OpenAI博客, 1(8):9, 2019年。
- [53] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharwan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu等。探索使用统一的文本到文本转换器进行迁移学习的极限。J. Mach. Learn. Res., 21 (140) : 1-67, 2020年。
- [54] Javier Rando, Daniel Paleka, David Lindner, Lennart Heim和Florian Tramèr。对稳定扩散安全过滤器进行红队测试。arXiv预印本 arXiv:2210.04610, 2022年。
- [55] David Rolnick, Andreas Veit, Serge Belongie和Nir Shavit。深度学习对大规模标签噪声具有鲁棒性。arXiv预印本 arXiv:1705.10694, 2017年。
- [56] Johann Schlamp, Josef Gustafsson, Matthias Wählisch, Thomas CSchmidt, and Georg Carle. 互联网的被遗弃的一面: 在域名过期时劫持互联网资源。在国际流量监测和分析研讨会上, 第188-201页。Springer, 2015年。
- [57] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman等。LAION-5B: 用于训练下一代图像-文本模型的开放大规模数据集。arXiv预印本 arXiv:2210.08402, 2022年。
- [58] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev 和Aran Komatsuaki。LAION-400M: 剪辑过滤的4亿个图像-文本对的开放数据集。arXiv预印本 arXiv:2111.02114, 2021年。
- [59] Christoph Schumann和Romain Beaumont. LAION-美学。<https://web.archive.org/web/20230119181400/https://laion.ai/blog/laion-aesthetics/>, 2022.
- [60] Roei Schuster, Congzheng Song, Eran Tromer和Vitaly Shmatikov。You autocomplete me: 毒害神经代码完成的漏洞。在第30届USENIX安全研讨会 (USENIX Security 21), 页码1559-1575, 2021年。
- [61] Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, 和Armand Joulin。CCMatrix: 在网络上挖掘数十亿高质量的平行句子。arXiv预印本arXiv:1911.04944, 2019年。
- [62] Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville和Yoshua Bengio。用循环神经网络生成事实问题: 3000万个事实问题-答案语料库。arXiv预印本arXiv:1603.06807, 2016年。
- [63] Pnina Shachaf和Noriko Hara。超越破坏行为: 维基百科的恶作剧者。信息科学杂志, 36 (3) : 357-370, 2010年。
- [64] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christopher Studer, Tudor Dumitras和Tom Goldstein。毒蛙! 针对神经网络的有针对性的干净标签投毒攻击。神经信息处理系统的进展, 2018年。
- [65] Piyush Sharma, Nan Ding, Sebastian Goodman和Radu Soricut。概念字幕: 用于自动图像字幕的清洁、上位词化的图像替代文本数据集。在计算语言学协会第56届年会 (第1卷: 长篇论文), 页码2556-2565, 2018年。
- [66] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Croux-iron, Wojciech Galuba, Marcus Rohrbach和Douwe Kiela。Flava: 一种基础的语言和视觉对齐模型。在 IEEE/CVF计算机视觉和模式识别会议的论文集中, 页码15638-15650, 2022年。
- [67] Johnny So, Najmeh Miramirkhani, Michael Ferdman, and Nick Nikforakis。域名确实会改变它们的特点: 量化潜在的滥用剩余信任。在IEEE安全与隐私研讨会上的论文集中, 页码为2130-2144, 2022年。
- [68] Lukas Struppek, Dominik Hintersdorf, Daniel Neider, and Kristian Kersting。学习打破深度感知哈希: 使用案例 NeuralHash。在2022年ACM公平性、问责性和透明度会议的论文集中, 页码为58-69, 2022年。
- [69] Besiki Stvilia, Michael B Twidale, Linda C Smith, and Les Gasser. Information quality work organization in Wikipedia. Journal of the American society for information science and technology, 59(6):983-1001, 2008.
- [70] Alon Talmor和Jonathan Berant。作为回答复杂问题的知识库的网络。arXiv预印本arXiv:1803.06643, 2018年。
- [71] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth和Li-Jia Li。YFCC100M: 多媒体研究中的新数据。ACM通信, 59(2): 64-73, 2016年。
- [72] Antonio Torralba, Rob Fergus和William T Freeman。8000万个小图像: 用于非参数对象和场景识别的大型数据集。IEEE模式分析和机器智能交易, 30(11): 1958-1970, 2008年。
- [73] Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey和Jens Lehmann。LC-QuAD: 用于知识图谱上复杂问题回答的语料库。在国际语义网会议, 页210-218。Springer, 2017年。
- [74] Alexander Turner, Dimitris Tsipras, and Aleksander Madry。清洁标签后门攻击。2018年。
- [75] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, 和Thomas Wolf。扩散器: 最先进的扩散模型。https://github.com/huggingface/diffusers/blob/8178c840f265d4bee91fe9cf9fdd6dfe091a720/src/diffusers/pipelines/stable_diffusion/safety_checker.py, 2022年访问于2023年2月7日。

- [76] Eric Wallace, Tony Z Zhao, Shi Feng, and Sameer Singh. 对NLP模型的隐蔽数据投毒攻击。arXiv预印本
arXiv:2010.12563, 2020年。
- [77] Angelina Wang, Alexander Liu, Ryan Zhang, Anat Kleiman, Leslie Kim, Dora Zhao, Iroha Shirai, Arvind Narayanan, and Olga Russakovsky. REVISE: 用于测量和减轻视觉数据集偏见的工具。国际计算机视觉杂志, 第1-21页, 2022年。
- [78] Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. CCNet: 从网络爬取的数据中提取高质量的单语数据集。在第12届语言资源和评估会议上, 页码4003-4012, 法国马赛, 2020年5月。欧洲语言资源协会。
- [79] 维基百科贡献者。维基百科的可靠性-维基百科, 自由百科全书, 2022年。[在线; 访问日期2022年7月21日]。
- [80] 维基百科贡献者。维基百科: 放心, 破坏吧, 2022年。[在线; 访问日期2022年12月5日]。
- [81] 杨毅, 易文涛, Christopher Meek。WikiQA: 一个开放领域问题回答的挑战数据集。在2015年自然语言处理实证方法会议上, 页码2013-2018, 2015年。
- [82] Scott Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 通过分阶段查询图生成进行语义解析: 带有知识库的问答。在A CL和AFNLP的第53届年会和第7届国际联合自然语言处理大会的论文集中, 2015年。
- [83] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 理解深度学习(仍然)需要重新思考泛化。ACM通信, 64 (3) : 107-115, 2021年。

附加图

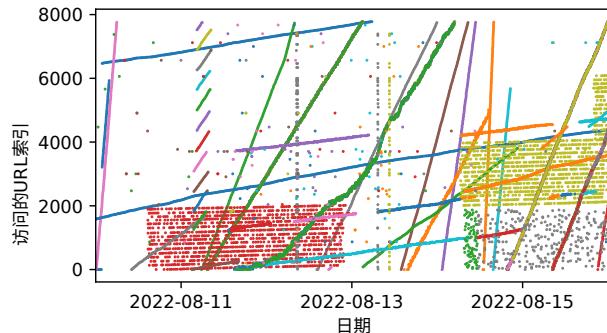


图8: 对包含在Conceptual 12M中的URL的服务器访问的未经过滤(没有任何精确度或召回率要求)的视图。与图2进行比较, 以获取经过滤的视图。

B 文本数据集的进一步讨论

在本节中, 我们进一步讨论文本数据集中存在的漏洞, 首先关注对维基百科的有针对性的投毒攻击, 然后关注Common Crawl数据集。

B.1 注释还原

在每种语言中, 我们生成一个常用于表示还原的单词列表。具体使用的单词是由每种语言的维基百科贡献者社区决定的, 因此没有具体的列表。然而, 在每种语言中, 都有自动还原工具和手动还原工具, 它们被标记为英文单词“reversion”或简称“rv”。这些是我们手动分析的起点: 我们确定大致翻译为“还原”、“撤销”或类似含义的单词, 并且这些单词也出现在我们确定为自动还原或手动还原的还原评论的样本中。然后, 我们使用每个新识别出的单词对评论进行抽样, 以验证它们是否捕捉到还原的新实例(即, 它们用于唯一标记语言的维基百科中的还原), 并且不会产生太多的误报。

总的来说, 我们不指望这个列表对于任何特定的语言都是完美的, 因为本文的作者没有参与任何语言维基百科的编辑。然而, 我们相信我们的分析足以验证我们注意到的两个趋势: 在非英语维基百科上仍然可能发生前置攻击, 并且在非英语维基百科上的攻击可能更加强大。

B.2 前置攻击的级联效应

我们的攻击使得对手能够污染任何维基百科快照。对手可以利用前置运行来损害直接依赖这些快照的下游数据集。与随意修改大部分维基百科不同, 这些攻击可以非常有针对性地进行。

这些攻击可以通过非常有针对性的前置运行投毒来实现。

- 知识库提取数据库在分析中被广泛依赖（例如，提出类似“哪些政治家是帮派成员？”的查询）和机器学习中（例如，用于训练和评估问答模型）。其中一个最大的数据库是DBPedia [2, 42]。由于该数据库是使用每月快照创建的，它们在所有意义上都是对维基百科的一种过滤视图。这使得对手能够直接利用我们的前置攻击以更有针对性的方式对这些数据库进行投毒。

作为分析的一个例子，用户可能发出这样的查询：“哪些政治家是帮派成员？”有针对性的投毒攻击可以强迫个人被错误地包含在这样的列表中，通过前置攻击进行小的编辑。更广泛地说，这样的攻击可以被设计成伤害特定个人或降低某些聚合统计数据的可靠性。

类似地，在机器学习的问答任务中，一些数据集，如LC-Quad [22, 73]，将数据存储为问题-查询对。攻击者还可以通过前置攻击来针对这些数据集来破坏查询。由于维基百科的快照不频繁，并且像这些知识库这样的下游系统可能更不频繁地更新它们的终端节点[2]，这些攻击可以在多个月内保持有效。另一种常见的方法是直接将数据存储为问题-答案对[9, 12, 62, 70, 82]。这可以减轻我们上述的攻击，前提是用于最初生成数据集的特定检查点没有被破坏。然而，这只能防止对在这些下游任务上训练的模型进行投毒 - 对在被投毒的维基百科快照上进行微调的模型仍然是脆弱的。最后，我们注意到这些攻击的影响在多语言数据上可能会加剧，正如我们在第5.1节中讨论的那样。

B.3 常见爬行

Common Crawl是一个以PB级规模的网络爬行数据集，每月大致重复捕获一次。每个存档都是对互联网的完整重新爬行，记录了爬虫的所有请求和主机响应，包括HTTP头和内容。因此，每个存档都包含了访问时的所有爬行页面的静态快照。这可能包括之前爬行中未见过的新页面内容，并且可能不包括自上次爬行以来已过时的内容。例如，2022年9月24日至10月8日期间爬取的数据包含了34百万个注册域名的380 TiB未压缩内容的31.5亿个网页，其中13亿个URL在之前的任何爬行中都没有被访问过。¹⁴Common Crawl数据集容易受到类似于我们的前置和分割视图投毒攻击的攻击。

¹⁴<https://commoncrawl.org/2022/10/sep-oct-2022-crawl-archive-now-available/>

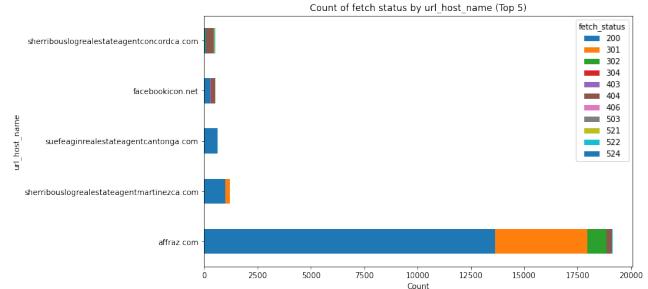


图9：按状态计数的前5个域名

攻击者可以购买一个过期的域名，该域名以前包含在Common Crawl中，然后使用攻击者选择的内容重新爬取该域名，然后该内容将出现在随后的Common Crawl快照中。请注意，与对维基百科的快照投毒攻击不同，这里没有内容审核，因此攻击者只需继续控制该域名即可投毒所有未来的Common Crawl快照。购买最近过期的存在于以前的Common Crawl快照中的域名，可以进行一种更强大的攻击形式，攻击者可以向爬取中注入全新的链接。这可以通过向被投毒的域名添加链接或子域名，并允许爬虫发现新的被投毒域名来实现。因此，攻击者可以向Common Crawl数据集中注入任意多的页面，不仅限于最初过期的子集。根据我们之前的道德声明，我们不实施这种攻击。

由于Common Crawl WARC文件已经由亚马逊在AWS Athena（无服务器服务）上托管¹⁵，因此分析URL的域名侦察工作是廉价的。扫描10年的Common Crawl数据以分析来自热门顶级域名和大量Common Crawl条目的域名对我们的成本仅为0.84美元。虽然额外的分析可能会稍微增加这个成本，但它仍然是一种廉价的搜索易受攻击的域名的方法。购买最近过期的域名，或者具有悬挂DNS记录和活动IP地址的域名是首选，因为在连续爬行中未能返回200-OK状态的域名似乎被移至较低优先级。例如，我们购买的过期域名中，仅有一个域名占所有购买域名的90%以上的状态码，而其他域名早在2020年12月20日就已经购买，但在3年的时间内看到的爬取流量相对较少。¹⁶由于Common Crawl非常庞大且未经筛选（以准确反映互联网的内容），毒化整个Common Crawl是不切实际的，因为其规模太大。此外，它

¹⁵<https://commoncrawl.org/2018/03/index-to-warc-files-and-urls-in-columnar-format/>

¹⁶本研究中使用的域名返回了404 HTTP状态，明确防止内容的抓取，可能影响域名再次出现的分析。也就是说，之前返回404的域名现在返回200。

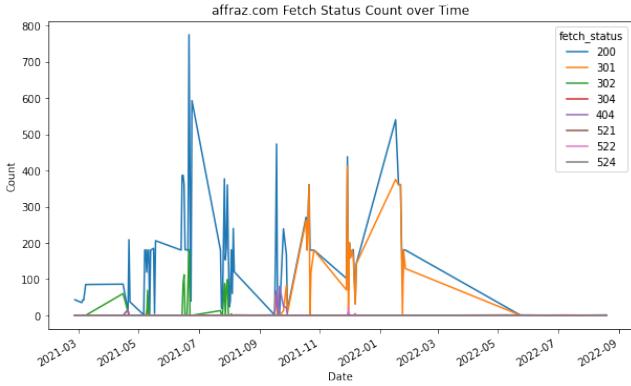


图10: affraz.com随时间的状态计数

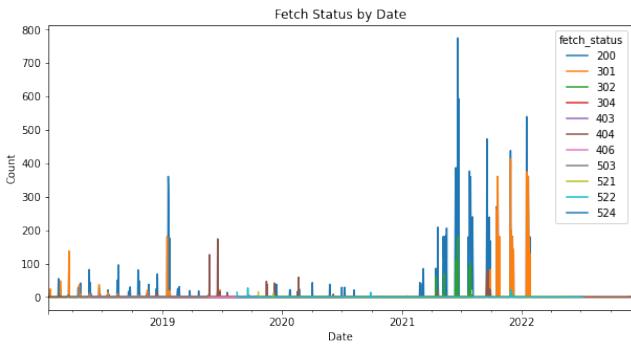


图11: 所有购买的域名的状态计数总和

消费者在处理这些数据用于下游的机器学习任务时，并不总是明显的。然而，存在许多派生数据集，这些数据集是通过对Common Crawl的相关子集进行筛选构建的。这包括LAION-5B图像数据集[57]，被称为Pile的文本数据集[23]，多语言文本数据集CC-100[78]和CCMatrix数据集[61]，以及一对翻译句子的翻译数据集。这种筛选实际上增强了攻击的威力：向Common Crawl添加1 MB的文本将污染Common Crawl的 $2.5 \cdot 10^{-9}$ 的部分，但如果这段文本绕过了CC-100数据集的筛选，它可能会污染英语语料库的 $1.2 \cdot 10^{-5}$ 的部分，甚至是整个Oromo语料库的9.1%。

C完整性检查防御的限制

在第6.2节中，我们概述了一种自然的防御方法，用于对抗分割视图投毒攻击，该方法在数据集索引中添加完整性检查。具体而言，在收集数据集时，维护者会计算每个图像的加密哈希值，并将该哈希值添加到索引中。在下载数据集的副本时，客户端下载器会丢弃哈希值不再匹配的图像。不幸的是，这种防御方法存在严重问题。

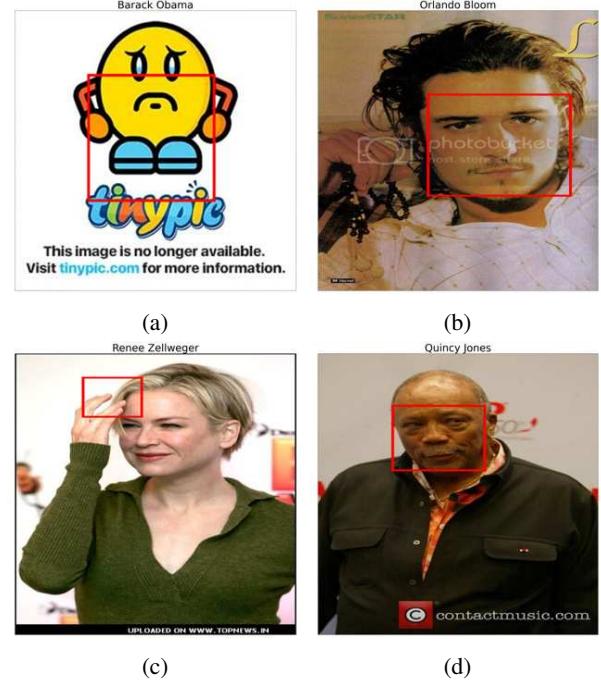


图12: 当图像不再与原始哈希值匹配时会发生什么？我们展示了来自PubFig数据集的4个示例。在图12a中，图像被替换为占位符。在图12b中，水印部分覆盖了面部，使该图像在训练中稍微不那么有用。在图12c中，图像被调整大小，因此原始边界框不再与面部匹配。与前面三个图像不同，图12d非常适合训练：水印位于面部边界框之外。

对效用的影响。正如我们在第6.2节中所看到的，为Conceptual Captions 3M数据集实施这种防御措施将导致丢弃大约55%的整个数据集，主要是因为诸如重新调整大小之类的微小但良性的图像变化。

在这里，我们对加密完整性保护对其他数据集的影响进行了更深入的分析。

C.1 野外修改图像：对PubFig数据集的案例研究

PubFig [34]是一个2010年的数据集，索引了近60,000张来自200个不同名人的图像，这些图像托管在各种领域中。PubFig是分布式数据集的最早例子之一。因此，它的索引中的许多URL可能已经不再有效（我们称之为“链接腐败”现象），而且剩下的URL中的许多可能指向随着时间的推移而略微修改的图像。由于小图像修改会丢弃许多完全有用的训练图像，因此完整性检查防御措施变得困难。在图12中，我们展示了原始图像有四种不同的方式

数据集	# 图像成功	下载无效		
		失败	图像	
pubfig	58795	35.2%	54.7%	10.2%
facescrub	106863	48.5%	44.5%	7.0%

表2：使用img2dataset[5]下载旧图像数据集时的链接失效。在旧的数据集中，不再在线上可用的图像比例很高。

准确率	准确率			准确率	数据集上的
	下载的修改后的、无裁剪的、修改后的	准确率	原始的、无裁剪的、原始的、裁剪的		
pubfig	20668	52.4%	59.7%	55.2%	96.0%
facescrub	51847	29.4%	93.3%	91.6%	97.7%

表3：成功下载的图像的变化量化。修改后的图像是指与原始图像具有不同哈希值的图像。我们将准确率定义为具有与任何可能标签接近的CLIP嵌入的修改后图像的比例，从而近似回答“图像中是否有可识别的人脸？”的问题。裁剪和未裁剪图像之间的差异主要是由于调整大小的图像使得裁剪无法捕捉到人物的脸部，如图12c所示。

在PubFig中，已经对其进行了修改。虽然我们找到了真正的正例，即图像确实被显著修改了，但也有很多情况下图像只经历了一些微小的变化，比如添加了水印。

许多图像数据集为图像的相关部分提供了边界框。PubFig也是如此：数据集索引包含了人脸边界框的坐标。如果客户端不重新计算这些边界框，任何图像的调整大小，比如图12c中的情况，都会使裁剪后的图像对训练毫无用处。

C.2 链接失效统计

基于加密哈希的完整性检查大大减少了某些数据集中可用的数据量。

我们下载了PubFig和FaceScrub数据集，并计算了“链接失效”的普遍性，即数据集索引中列出的URL不再有效或返回非有效图像。

除了死链接外，我们发现许多成功下载的图像已经被修改。我们在表3中展示了聚合统计数据。在PubFig数据集中，超过50%的存活图像存在哈希不匹配的情况。然而，这些图像中的许多是以无害的方式进行修改的。为了证明这一点，我们采用了一个预训练的人脸嵌入模型，并使用只有正确哈希的图像对PubFig类进行微调分类器。然后，我们在具有错误哈希的下载图像上评估这个分类器，发现我们只能达到46.8%的准确率。因此，这些图像中的许多被修改的方式使人物的身份变得模糊不清。

对于FaceScrub数据集，我们发现被修改的图像较少。成功下载的图像中约有70%仍与原始哈希匹配。在被修改的图像中，我们仍然可以正确分类88.2%，这表明大部分这些变化是良性的，如图12b或图12d所示。

对于PubFig，我们进一步调查了图像的频率

更改是否对应于良性调整，尽管如此使原始边界框变得过时（如图12c所示）。

为此，我们将CLIP (ViT-B-32-quickgelu来自[28]) 嵌入与数据集中200个公众人物的名称进行比较，阈值余弦相似度为0.21。我们发现，59.4%的具有错误哈希值的图像与CLIP空间中的任何标签都不接近。与表3进行比较，这相当于约4%的错误哈希值是类似于图12c的修改的结果。

感知哈希或类似方法可能缓解此问题，因为图像通常以微不足道的方式进行修改。

例如，如表1所述，COYO-700M图像

使用pHash [32]进行分发，该方法对良性

图像更改具有鲁棒性。然而，感知哈希不具备与密码哈希相同的最坏情况完整性保证

[29]。由于

错误使用感知哈希作为预像抗性

算法而引起了高调争议

[68]。人们普遍认为SHA-256的预像攻击是不可行的，而没有已知的感知哈希函数具有类似的安全保证。

D LAION 攻击细节

第4.5节中的两种攻击方法会对CLIP模型进行投毒，使得一些固定图像的嵌入接近目标文本标签的嵌入。我们实验中的关键技术限制是所有攻击都需要并行进行，以最小化成本，因为重新训练CLIP是非常昂贵的。

目标误分类目标。ImageNet数据集包含1000个图像类别。ImageNet上的CLIP零样本分类器返回与图像嵌入在CLIP潜空间中的文本嵌入具有最大余弦相似度的类标签。

我们的投毒攻击目标是使 CLIP 零样本分类器将特定图像分类为错误的标签。

我们选择了10个类别作为投毒的目标标签，这些标签在与廉价可购买域名相关的标题中至少出现1000次；请参见表1。对于每个选择的标签，例如苹果：从可购买域中选择1000个标题-图像对 S 苹果，其中标题中包含苹果。我们还要求所有选择的类别的跨越 S 类别的域名的总成本不超过1000美元。然后，我们选择一张单独的不相关的图像 I —通常不会被分类为苹果—并将 S 苹果中的1000张图像替换为图像 I 。因此，对于这10个类别中的每一个，我们投毒了1000张图像，仅占数据的0.000025%。

NSFW目标。这种攻击的目标是使得随附于Hugging Face扩散器库[75]中的稳定扩散1.4模型的**NSFW**过滤器将给定的良性图像错误标记为**NSFW**。分类器是在CLIP潜在空间中的余弦相似度阈值函数，将图像嵌入与文本**NSFW**概念列表[54]进行比较。

我们选择了10个良性图像，并对每个图像 I 执行以下操作：从可购买域中选择1000个标题-图像对，使得标题在LAION 400M元数据中被标记为不安全，并将对应的1000个图像分别替换为 I 。再次，我们选择了总成本低于1000美元的域中的图像。实验。对于两种攻击（以及所有选择的图像），我们同时在LAION 400M

数据集上使用所描述的修改训练了一个OpenCLIP [28]模型。我们使用16个A100 GPU，在批量大小为3072的情况下，对ViT-B-32 CLIP模型进行了32个时期的训练。目标误分类攻击对于60%的目标有效：选择的图像被零样本CLIP分类器分类为目标标签。NSFW攻击对于90%的目标图像有效。

购买域名的落地页

以下文本被放置在我们购买的每个域名的落地页上。

该域名是研究项目的一部分。该域名是作为一个研究项目的一部分购买的，研究机器学习数据集随时间变化的程度。

该域名曾包含在其中一个数据集中，并托管了该数据集的一部分图像，但之前的所有者让该域名过期了。我们于2022年8月购买了这个域名，以衡量从这些过期域名查询的人数。

我们购买了许多包含在不同数据集中的域名。除了首页之外，所有这些域名都将返回404错误。您无需采取任何额外措施来确保您的数据集不受我们的研究影响：如果我们没有购买这个域名，URL将为NXDOMAIN，您将不会收到任何内容。

我们可能会暂时记录发送到该服务器的请求的元数据，以衡量对该域名的抓取行为的普遍程度。

我们记录的任何数据将在研究完成后被删除。如果您不愿意参与这项研究，请通过下面的电子邮件地址与我们联系，我们将删除您可能为我们的研究做出的任何贡献的数据。

如果您允许我们使用您的数据，我们将非常感激；我们认为这将是一项有价值的研究。

如果您对这项研究有任何问题，可以通过dataset-expired-domain-study@googlegroups.com与我们联系。获取更多信息。

这曾经是我的域名。我要回来吗？如果您是这个域名的原始所有者，我们将很乐意根据您的请求将其归还给您，通过上述电子邮件地址与我们联系。我们将在完成研究后让这个域名过期。

这会对我下载的数据集造成问题吗？

正如我们上面所说，如果您正在抓取这个数据集，您不需要采取任何措施来特别避免这个域名（或者我们购买的任何其他域名）。我们已经测试过，我们发送的404响应将导致所有标准图像下载工具完全跳过该图像。

你的研究会被发表吗？我们将在完成后发布我们的研究。我们预计这将在接下来的几个月内发生。如果您想要一份这项研究的副本，请与我们联系。

我有一个未提及的问题。请通过dataset-expired-domain-study@googlegroups.com与我们联系以获取更多信息。