CenRL: A Framework for Performing Intelligent Censorship Measurements

Hieu Le*, Armin Huremagic*, Kevin Wang*, Roya Ensafi*, Ram Sundara Raman†
*University of Michigan †University of California, Santa Cruz

Abstract—**Active Internet measurements are crucial for exposing the increasing frequency and severity of global Internet censorship. However, current measurement efforts are constrained by limited resources and time, and reacting to new censorship events remains a largely manual process reliant on rapid signals. As a result, creating a comprehensive and real-time picture of Internet censorship remains a key challenge. In this work, we introduce CenRL, an intelligent censorship measurement framework that leverages reinforcement learning to optimize and automate censorship measurements. We model the censorship measurement process as a multi-armed bandit problem and design CenRL agents to address two key tasks: maximizing the detection of blocked websites within a network and automatically responding to blocking changes in dynamic censorship environments. We demonstrate CenRL's effectiveness through realistic simulated experiments in three highly censored regions (China, Russia, and Kazakhstan) and real-world censorship measurements across vantage points in 15 countries with diverse censorship policies. Our controlled experiments demonstrate that CenRL significantly outperforms the state of the art measurement processes, finding 75% of blocked websites in less than half the number of measurements while identifying censorship changes up to seven times faster. Our real-world experiments confirm this advantage across multiple censorship environments, showing that CenRL can find 2.5 times more blocked websites on average compared to existing measurement strategies. Our study demonstrates the potential of using reinforcement learning to provide deeper insights into restrictions on online freedom.**

1. Introduction

Internet censorship is expanding rapidly, with governments and organizations increasingly deploying sophisticated methods to monitor and block online content. For example, the Freedom House recently reported growing online content restrictions in 21 countries [1]. In response, the academic and Internet freedom communities face the need to systematically collect data to record the escalating frequency and intensity of censorship events worldwide. To date, most censorship measurement efforts have primarily focused on developing highly accurate detection tools and applying them to study censorship in specific regions such as China, Iran, and Russia [2]–[5]. Notably, there exist longitudinal censorship measurement platforms like OONI [6] and Censored Planet [7] that test hand-curated lists of websites and provide the results as publicly accessible datasets, which have proven crucial to understanding Internet censorship.

Despite these efforts, challenges remain in studying the evolving censorship landscape comprehensively. First, most censorship measurement studies and platforms like OONI and Censored Planet are operated by non-profit organizations, networks of volunteers, or academic research labs. They face significant resource limitations, including limited network bandwidth, computing power, and viable vantage points. These limitations hinder the ability to scale measurements and characterize censorship policies thoroughly. Moreover, censorship measurement is guided by many ethical safety considerations, further limiting the speed or quantity of measurements [8]–[10]. As a result, most previous studies [6], [7] have been limited to testing a small set of politically sensitive [11] or extremely popular websites [12]. Second, studying censorship events as they occur in real-time remains primarily a manual process. Events like the HTTPS interception in Kazakhstan [13] and the increased blocking in Iran following the Mahsa Amini protests [14], [15] often lead to an expansion in the scale of censorship or cause alterations in the censorship mechanisms themselves. In a typical real-time scenario, Internet users on the ground alert researchers to these changes, who then orchestrate targeted measurements to study the event. Thus, this process is dependent on rapid reports and manual measurement orchestration, both causing delays in reporting. Ultimately, this results in incomplete gathering of data during critical periods.

In this study, we address these challenges by introducing **CenRL**, a reinforcement learning (RL) framework that optimizes and automates Internet censorship measurements through dynamic decision-making, efficiently utilizing the limited measurement resources available and rapidly adapting to the evolving landscape of censorship events. Inspired by the success of RL in adjacent fields such as network traffic classification and routing [16]–[21], we design CenRL to intelligently select and measure potential censorship targets through sequential decision making. At the core of CenRL is a novel formulation of the censorship measurement task as a multi-armed bandit (MAB) problem, where an intelligent entity, the CenRL agent, is given the goal of optimizing censorship detection within a limited time period. To do so, the agent follows an RL policy to select an action (e.g., a website to test) that balances the exploration of large action spaces (e.g., the Tranco Top–10K list of websites) with the exploitation of actions that are more likely to reveal censorship (e.g., websites with the same parent entity as a known blocked website). The agent receives a reward for applying the action and achieves its goal by maximizing the cumulative reward over the limited time period.

We design CenRL agents to address two key tasks in censorship measurement: (*Task 1*) maximizing the discovery of blocked websites within a network and (*Task 2*) rapidly and automatically detecting changes in blocking over time within a dynamic environment. For both tasks, CenRL operates on a large input list of websites to test, such as the Tranco list of popular websites [12]. We build CenRL's action space by utilizing website features—such as its subdomain, TLD, rank, category, and parent entity—as arms in the MAB framework. These features are carefully selected to capture censorship patterns, informed by our domain expertise and blocking trends observed in prior research, which show that censorship policies often target websites sharing some *similarity* [2], [4], [6], [7], [13]. For instance, Phong et al. report that the Great Firewall started to block many COVID-19-related websites together in 2020 [4]. Building on such insights, we design CenRL's action space and reward functions to prioritize testing websites with similar characteristics once a blocked website is identified, ensuring a more efficient measurement process.

Evaluating the efficacy of an RL-based measurement framework like CenRL is nontrivial, involving thousands of tests across regions and networks, hyperparameters, and design choices for RL components such as reward functions and action spaces. We address these challenges fully for CenRL by developing a practical, end-to-end pipeline and implementation for CenRL and evaluating it through both controlled experiments and real-world censorship measurements. First, to comprehensively train, tune, and evaluate CenRL, we design and implement three realistic, controlled censorship environments inspired by previous work on three regions: Russia [2], China [4], and Kazakhstan [13]. The simulations for Russia and China involve environments with a large number of blocked websites (200K+), while the Kazakhstan simulation models a specific event. We evaluate the performance of CenRL against baselines that represent both the current state of the art in censorship measurement and heuristic-based selection strategies. Second, we demonstrate that CenRL performs well in real-world settings by integrating it into an implementation of Hyperquack [22], a remote measurement technique employed by the Censored Planet platform [7]. Using this integration, we expand the scope of our evaluation to 15 countries in vantage points with varying levels of network restrictions, demonstrating the broad applicability of CenRL.

Evaluations from our three controlled experiments show that CenRL significantly outperforms current state of the art censorship measurement strategies. CenRL identifies over 75% of the blocked websites across all three controlled environments using fewer than half the measurements required by traditional methods, thereby saving time and resources. Moreover, when simulating longitudinal measurements in dynamic environments, CenRL automatically adapts to changes over time, identifying up to 250% more blocked websites. In the Kazakhstan dynamic environment, CenRL automatically responds to the censorship event, discovering newly blocked websites approximately seven times faster than existing methods by the second day of the event. Our real-world experiments further demonstrate the benefits of intelligent censorship measurement, showing that CenRL enables tools like Hyperquack to detect substantially more instances of blocking in nearly every country tested, yielding an average increase of 2.75× in the number of blocked websites identified. Moreover, in addition to detecting more blocked websites, CenRL also identifies blocked websites *faster* than current approaches. By training on historical data and making decisions dynamically, CenRL identifies and exploits blocking patterns across diverse censorship contexts.

Our experiments demonstrate that the CenRL framework can significantly enhance censorship measurements and adapt to diverse censorship landscapes across regions and over time. We open-source CenRL[1], which contains an API that can be used by active censorship measurement platforms to integrate with the framework and query intelligent, real-time measurement inputs for their longitudinal operations. Our work represents a transformative step towards equipping the censorship measurement community with intelligent tools that can keep pace with evolving network restrictions.

2. Background & Related Work

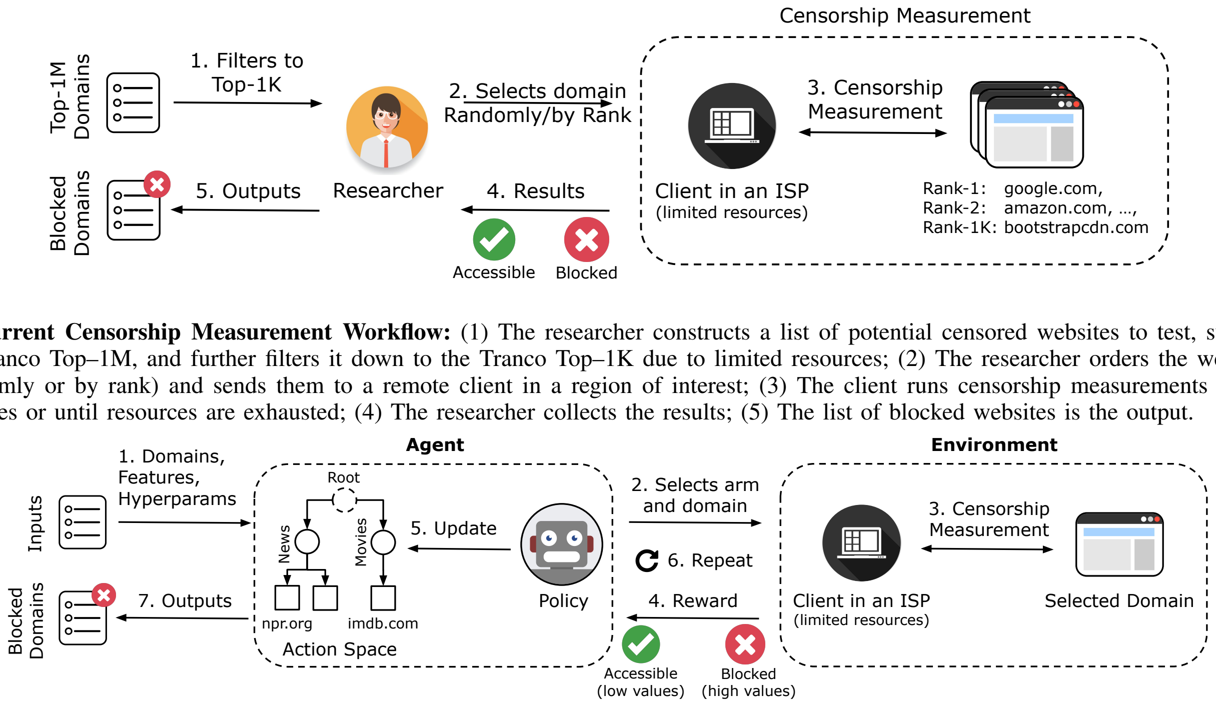
Censorship. A *censor* inspects Internet traffic and disrupts (e.g., using a TCP RST packet) or drops connections that contain forbidden content or targets. The blocking might occur during different stages of a network connection. For instance, a censor might interfere with the DNS resolution process by either blocking the client from obtaining an IP address or supplying an incorrect IP address [4], [23]–[25]. A censor might also block a client from establishing a transport-layer (e.g., TCP) or application-layer (e.g., HTTP(S), FTP) connection with a server by dropping or injecting packets [2], [13], [22], [26]–[28]. Extensive research has shown the diverse patterns of Internet censorship around the world [4], [6], [7], [15].

2.1. Censorship Measurement Platforms

The growing prevalence of censorship across services and regions has heightened the demand for collecting and analyzing data to understand what is being censored. Addressing this need, researchers and non-profits have developed and deployed multiple censorship measurement platforms [4], [6], [7], [29]–[31]. The following are the largest active censorship measurement platforms with longitudinal data on website censorship, which is our focus in this work.

Censored Planet. The Censored Planet platform specializes in remote measurements to public infrastructural machines on the Internet (e.g., routers, open DNS resolvers, and web servers) and infers censorship based on responses received

[1] https://github.com/r-andlab/cenrl, DOI: 10.5281/zenodo.17382167



(a) **Current Censorship Measurement Workflow:** (1) The researcher constructs a list of potential censored websites to test, such as the Tranco Top–1M, and further filters it down to the Tranco Top–1K due to limited resources; (2) The researcher orders the websites (randomly or by rank) and sends them to a remote client in a region of interest; (3) The client runs censorship measurements for all websites or until resources are exhausted; (4) The researcher collects the results; (5) The list of blocked websites is the output.

(b) **CenRL Workflow:** (1) The researcher provides inputs, such as the Tranco Top–1M websites, features (website categories) to construct the action space, and RL hyperparameters; CenRL initializes the workflow, such as building the action space; (2) The agent follows the RL policy to select an arm and corresponding website to test and sends it to the environment; (3) The client in the environment runs the censorship measurements for the selected website; (4) The environment returns a reward (e.g., higher rewards if the website was determined to be blocked); (5) The agent uses the reward to update the expected value of the arm; (6) CenRL repeats 2–5 for a time horizon T; (7) The list of blocked websites is the output.

Figure 1: **Current Censorship Measurements vs. CenRL Workflow:** Instead of selecting measurement targets using manual heuristics, CenRL leverages learning to use limited resources efficiently while optimizing censorship discovery.

from these machines [7], [22], [24], [27], [32]. Censored Planet longitudinally tests the reachability of around 2,000 popular [12] and sensitive [11] websites from more than 90,000 remote vantage points [33].

OONI. The Open Observatory of Network Interference uses the OONI probe to collect direct censorship measurements from volunteer devices [6]. Over the past decade, the OONI project has built a global community of volunteers that contribute measurements using hand-curated test lists [11]. Within its static set of potential test websites, OONI prioritizes the testing of certain websites by manually assigning a higher weight to certain public-interest website categories. We further discuss OONI's selection process and contrast it to CenRL's approach in Appendix A.

GFWatch and GFWeb. These platforms collect longitudinal data regarding the blocking behavior of the Great Firewall in China [4], [30]. They specialize in collecting targeted blocking behavior regarding a large number of websites. For example, GFWatch has detected more than 669,000 blocked domains in China since its launch in 2020 [34].

CenRL's Goal. The goal of our work is to explore, build, and evaluate reinforcement learning methods that can enhance the operation of these censorship measurement platforms. We are guided by the intuition that past measurement results should enhance future measurements. We describe CenRL's integration into an implementation of Censored Planet's Hyperquack measurements in §5.2.

2.1.1. Workflows and Challenges. Fig. 1a depicts the typical high-level workflow of a censorship measurement platform. Researchers are tasked with identifying which websites are blocked within a network. Typically, they do not have prior knowledge about blocking. They decide upon a list of potential websites to test, such as the Tranco Top–1K list of popular websites [12], depending on limited resource availability. Without prior knowledge, the current state of the art selection approach is to randomly select websites to test within the list or order them by popularity rank [4], [7]. The researchers then conduct censorship measurements for the list of websites from a client connected to the network of interest until resources are exhausted or a certain time limit is reached (e.g., typically 90 seconds in OONI). For example, to detect SNI-blocking, the client may send TLS Client Hello handshake messages for each domain and observe the responses for signs of blocking. The results are returned to the researchers, and the above process is often repeated for longitudinal measurements.

Real-world Challenges. The workflow above captures valuable censorship data. However, performing censorship measurements at scale repeatedly encompasses several challenges. First, obtaining and maintaining access to hosts or volunteer devices within censored networks is expensive and difficult due to network and legal restrictions, especially when performing high-frequency measurements that utilize large bandwidth (several Gbps) for extensive time periods [4], [6], [29]. Second, while remote measurements [7], [22], [24], [27], [32], [35] offer a cost-effective alternative to direct network access, they are required to be heavily rate-limited to avoid service disruptions to remote endpoints. For instance, Censored Planet is only able to measure access to around 2,000 websites over the course of a week [7], [33]. Third, censorship measurements carry legal and technical risks for hosts and endpoints, requiring careful thought and design in measurement safety. This leads researchers to either obtain informed consent from the user providing the host or, in case of remote measurements, further rate-limit measurements and select remote endpoints carefully. While some measurement studies have successfully studied access to millions of domains [2], [4], [30], [35], [36], all such studies focus on specific regions, use specialized, limited infrastructure or measurement techniques, and do not make measurement decisions adaptively. Finally, there are *dynamics* to censorship mechanisms and policies: they evolve quickly during momentous events [13], [14], [37]–[39]. Current studies require both information from on-the-ground contacts as well as manual efforts in changing measurement configurations to capture such changes. This leads to sub-optimal characterization of many events.

In this work, we address these challenges by developing an approach that enables censorship measurement platforms to utilize their resources optimally and dynamically. Guided by the typical measurement workflow depicted in Fig.1a, we identify a critical missing step: the outcome of whether a website is blocked should inform researchers' subsequent decisions when selecting the next websites to test. Similarly, in longitudinal measurements, an observed change in blocking can inform researchers about how the environment has evolved. In §3, we show that this problem can be effectively framed as an RL process.

2.2. ML in Censorship Research

Machine learning has been widely used by anomaly-based intrusion detection systems to distinguish suspicious packets from benign traffic flows [16], [17], [40]–[43] and to classify internet traffic and its routing [18], [19], [44], [45]. Some studies have explored the application of ML in the censorship research space. Bock et al. developed Geneva, a tool that uses genetic algorithms to generate new species of traffic obfuscation to evade censors [46]. Zhu et al. developed an unsupervised learning approach to detect adversarial packets that can elude DPI middleboxes [47]. Calle et al. and Brown et al. applied supervised and unsupervised ML models to identify anomalies in DNS measurements collected by OONI [48], [49]. Tsai et al. modeled Censored Planet data as decision tree structures in order to identify and characterize censorship events [50]. While these previous work use ML for censorship circumvention or data mining, CenRL focuses on *censorship measurement*.

We model the censorship measurement environment as a multi-armed bandit (MAB) problem [51]. MAB problems, originally conceptualized in the context of slot machines, present a suitable analogy for networking and security scenarios, where limited resources must be allocated efficiently to optimize specific objectives. For example, MAB models have been applied to address resource allocation [52], network routing and scanning [18]–[20], [53], and access point selection [54]. In security research, MABs have been leveraged for vulnerability fuzzing [55], [56], intrusion detection [57], and generating filter rules for adblocking [21].

3. CenRL Framework

We present CenRL, an RL framework that transforms the workflow of censorship measurements from Fig. 1a to Fig. 1b and optimizes two common tasks—*Task 1:* The discovery of a comprehensive list of blocked websites, and *Task 2:* Automatically adapting to changes in blocking in a dynamic censorship environment. Our goal is to optimize these tasks while addressing the challenges of limited resources and constantly evolving environments.

3.1. Formulation Overview

Multi-armed bandit. We formulate the censorship measurement process as an MAB problem. In the traditional context, an RL agent is faced with K arms with unknown and independent reward distributions, and it must learn through experience the best arm to maximize its cumulative reward R over a time horizon T. The MAB problem revolves around exploring this set of K arms and their reward distributions, denoted as $D_1, ..., D_K$, and characterized by their expected values $(Q_1, ..., Q_K)$ and variances $(\sigma_1^2, ..., \sigma_K^2)$.

During each time step, $t = 1, 2, \dots, T$, the agent follows a policy π to select an arm (indexed by a_t) and then apply it upon an environment. The environment returns a reward based on the selected arm's reward distribution (reward r_t drawn from D_{a_t}). The policy enables the agent to balance the exploration (i.e., unknown arms that may have high rewards) vs. exploitation (i.e., arms that are known to have high rewards) of an action space $\mathcal{A}$ constructed from the set of K arms. By the end of T, the agent has good estimates of the expected rewards of each arm.

Censorship Measurement as an MAB problem. As illustrated in Fig. 1b, an RL agent is introduced into the decision-making process. The agent follows a *policy* π to select an arm and then apply it to an environment. The *action space* is the set of arms containing all the content to be tested. We consider an arm to be a grouping of test websites, i.e., selecting an arm also involves selecting a website to test. The *environment* encompasses the capabilities of the measurement technique and the region where the test website is potentially censored. We consider both static environments, where censorship does not change over time (only *Task 1*) and dynamic environments, where censorship changes over time (*Tasks 1 and 2*). When the agent tests the selected website, it represents running a censorship measurement to determine whether it is blocked. The result is utilized to calculate a *reward* that represents the researcher's objective; e.g., high rewards for discovering censored content and low rewards otherwise.

From the extensive literature of RL, we choose MAB as our problem formulation for several reasons. First, the testing and reward of a selected arm are independent of other arms, i.e., when multiple websites are measured in parallel, they do not affect each other's outcome or rewards. Second, there are inherent dynamics to our arms: (1) there may be different outcomes when testing different websites within the same arm; and (2) the rewards of testing websites themselves can change over time. Thus, an arm must be tested multiple times to learn its expected reward. Third, the order of testing arms and websites does not affect their rewards. As a result, RL approaches that consider the concept of a "state" where the order of actions often matters, such as Q-learning, are not appropriate for our problem.

3.2. Action Space

Test Content i: is an input to CenRL in a set $\mathcal{I}$. For instance, this can be a website (i) from the Tranco Top–1M list ($\mathcal{I}$).

Action Space $\mathcal{A}$ and Arm(s) a: An action space $\mathcal{A}$ is constructed from $\mathcal{I}$. It consists of arms, where each arm a is a grouping of i, $a = \{i_1, i_2, \ldots, i_n\}$. For example, arm a can be a grouping of websites based on the website's parent entity. Thus, $\forall i \in \mathcal{I}$, we construct the action space such that $i \in \mathcal{A}$. Arms are independent from each other.

Constructing Arms: How we construct arms is crucial to achieving better results with CenRL. In the naive case, we can treat every a as each independent i, i.e., $|a| = 1, \forall a \in \mathcal{A}$ — this gives suboptimal results because once an arm is selected and tested, it will not be selected again. This defeats the purpose of our formulation, as each action does not inform subsequent actions. In §4.1, we outline how we construct arms based on natural groupings of test websites.

3.3. Reward Function

We design reward functions guided by the common goals of censorship measurement platforms. We utilize *Test(i)* to represent a censorship measurement that determines whether a test website i is blocked ($\mathcal{B}$).

Task 1: The primary goal is to maximize the discovery of censored content, such as websites. For a given time t, chosen action a, we select a potential censored content $i \in a$ and apply censorship measurement on i. We define the base reward function $R_1(a, i)$ as:

$$R_1(a,i) = \begin{cases} 1, & \text{if } Test(i) \text{ returns blocked } (\mathcal{B}) \\ 0, & \text{otherwise } (\neg\mathcal{B}) \end{cases} \tag{1}$$

Eq.(1) observes the response for a measurement and assigns a reward of 1 if blocking is observed for the chosen action. This reward values arms that contain censored websites.

Task 2: Another common goal for censorship measurement research is to automatically discover changes in censorship policies during censorship events, which can help in timely data-driven advocacy [13], [14], [37], [50], [58]. To do so, we design a reward function that tracks whether a selected test website was blocked on the previous day. Recall that in the real-world setting, a censorship measurement platform can conduct a limited number of measurements per day. To transform a time step t to a given day, we utilize a window of w to denote the number of time steps or measurements per day, i.e., a the number of measurements per day $d = \lfloor t/w \rfloor$.

Thus, for a given time t, we select an action a and test content $i \in a$. Here, we track whether i was blocked from the previous day, $d - 1$, recorded as $\mathcal{B}_{d-1}$ for blocked and $\neg\mathcal{B}_{d-1}$ for not blocked. Specifically, we define this reward function $R_2(a, i)$ as:

$$R_2(a,i) = \begin{cases} 1, & \text{if } Test(i) \wedge \neg\mathcal{B}_{d-1} \\ \frac{3}{4}, & \text{if } Test(i) \wedge \mathcal{B}_{d-1} \\ \frac{1}{2}, & \text{if } \neg Test(i) \wedge \mathcal{B}_{d-1} \\ 0, & \text{if } \neg Test(i) \wedge \neg\mathcal{B}_{d-1} \end{cases} \tag{2}$$

Similar to Eq.(1), Eq.(2) values the discovery of censored websites the most, denoted by the first two cases. We provide a higher positive reward (1) for finding new blocking as compared to finding existing blocking ($\frac{3}{4}$), prioritizing the discovery of changes. Moreover, for the third case, we also provide a positive reward ($\frac{1}{2}$) when i switches from blocked to not blocked, which is also of interest to the community [58].

3.4. Reinforcement Learning Policy

Given an action space $\mathcal{A}$, the agent selects arms sequentially, aiming to maximize cumulative reward over T steps. The agent follows a policy π that balances exploiting arms with known high rewards and exploring unknown arms that might yield higher rewards. We explore policies from three well-known algorithms for MAB problems: Upper Confidence Bound (UCB) [51], [59]–[61], ϵ-greedy Sampling [62]–[64], and Thompson Sampling [65], [66]. Both UCB and ϵ-greedy track an arm's expected reward using a moving weighted average: $Q_{t+1}(a) = Q_t(a) + \alpha(r_t - Q_t(a))$, where $Q_t(a)$ is the current expected reward of arm a, and α is the learning step size, a hyperparameter that adjusts how much the agent updates its reward expectations. Thompson sampling uses a β-distribution to adjust expected rewards. When there is no prior knowledge of the expected reward for arms, the initial expected reward $Q_1(a) = 0, \forall a$. When conducting our real-world experiments, we follow

Feature	Arms	# per arm	Examples (Feature → Website)
Category	640	16 ± 81	Technology → `amazonaws.com`
TLD	435	23 ± 274	com → `google.com`
Subdomain	1,393	2 ± 14	m → `m.facebook.com`
Rank bin	50	200 ± 0	Rank 400-600 → `weibo.com`
Entity	3,056	3 ± 17	Cloudflare → `cloudflare.com`

TABLE 1: **Action Space Features:** We characterize the action space per feature based on the number of arms and websites per arm created by the Tranco Top–10K websites. Here, # websites per arm shows the overall mean ± std.

an end-to-end pipeline where we train and tune the initial expected rewards ($Q_1(a)$) using historical data from Censored Planet. Since we find similar results using all three policies, *we only describe the UCB policy and its results throughout the main body of the paper*; a description of ϵ-greedy sampling and Thompson sampling can be found in Appendix C, along with their evaluations.

The UCB algorithm encourages the exploration of arms with uncertain estimates of their expected rewards while exploiting known arms that give high rewards [21], [51], [55], [59]–[61]. It introduces an upper confidence bound, $U_t(a)$, that measures the uncertainty about the current expected reward. Specifically, $U_t(a) = c.\sqrt{\frac{log(t)}{N[a]}})$, where t is the number of actions performed so far, $N[a]$ is the number of times the arm a has been selected and tested, and c is a hyperparameter that controls the amount of exploration. Thus, at a given time t, UCB selects arm a_t such that $a_t = argmax_a[Q_t(a) + U_t(a)]$.

4. CenRL Implementation

In this section, we introduce our implementation of CenRL guided by our formulation in §3.

4.1. Action Space Implementation

We design each arm in CenRL to represent a group of websites to test, organized and grouped by various features. These features are carefully selected based on our domain knowledge of censorship patterns and insights from prior work [4], [6], [7], [29]. We construct our action space from a given set of websites (primarily the Tranco Top–10K) and implement it as a directed graph using a well-known Python graphing module, NetworkX. The action space is the primary consumer of memory and update time in CenRL, and our graph based design limits the memory requirement to less than six megabytes and adds negligible update time to measurements. We create arms using the following methods (as summarized in Table 1):

1. Website Category. A website category is a classification that describes and characterizes the content of a website. For example, `www.cnn.com` would be classified under the category "News & Media." Website categories are intuitively a good feature for censorship measurements since censorship policies are often created based on the categories or content of the websites themselves. We determine website categories by using Cloudflare's Domain Intelligence API [67] and the Citizen Lab Test List [11], which have been used by previous work [6], [7], [15], [68]. In total, we construct 640 arms for the category using the Tranco Top–10K websites, with an average of 16 websites within each arm.

2. Top-level Domain (TLD). We extract the TLD of the website, with the insight that some blocking policies target specific TLDs (such as country-specific TLDs). We identify 435 unique TLDs within the Tranco Top–10K.

3. Subdomains. We extract the subdomains of the website URL. There are 1,393 subdomains in the Tranco Top–10K, with each arm containing only a mean of two domains. While this could potentially limit how well the agent learns across arms, we show in our evaluations (ref. §6.1.1) that the subdomain feature performs surprisingly well, as common subdomains (e.g., `app`, `m`) are frequently targeted.

4. Rank Bin. We bin website ranks from the Tranco ranking [12] into groups of 200. Websites with similar popularity will be grouped under the same arm, which can reveal patterns regarding the blocking of popular websites. We select a bin size of 200 as it provides a good balance between the number of arms and websites within each arm.

5. Entity. The entity is the parent organization that owns the website. We retrieve this information using various approaches and sources, including WHOIS (registrant, admin, or tech organization), SSL Certificate (subject's organization), DuckDuckGo's Tracker Radar [69], and DisconnectMe's tracking protection list [70]. We use these sources sequentially, as the parent entity information is not always available in a single datasource. The entity feature results in an action space with the highest number of arms (3,056) since many distinct parent organizations exist. We show later in §6 that this feature is particularly useful when the agent is allowed to learn over a large number of measurements.

We note that CenRL is modular, which allows for easy extension of other features to its action space. Moreover, while we explore every feature individually in this paper, they can be hierarchically combined for finer-grained control. For instance, a hierarchical action space can produce a social media website in the top 200 rank bin.

4.2. Controlled Environment

As discussed in §2.1, performing censorship measurements at scale is constrained by infrastructural, legal, and ethical considerations. To overcome these challenges and conduct the necessary extensive evaluation of CenRL, we create controlled but realistic environments using three existing censorship blocklists, described in Table 2: (1) Hoang et al. investigated DNS-based blocking in China and continuously publish an extensive list of blocking rules on the GFWatch platform [4], [34] (referred to in this work as `CN`); (2) Ramesh et al. studied censorship in Russia using a leaked authoritative blocklist, which is continuously updated on Github [2], [71] (referred to in this work as `RU`); (3) Sundara Raman et al. published a list of 37 domains undergoing

	Dynamic Environment			*Static Environment*			
List Name	**Date Range**	**Rules**	**Top–10K**	**Date**	**Rules**	**Top–10K**	**Examples**

`CN` [4]	2023-05-25
	(+30)
		282.2K ± 13.6K	1.05K ± 11	2023-12-16	222.2K	1,060	`*.google.com,`
							`*.facebook.com`

`RU` [2]	2023-11-01
	(+30)
		384.2K ± 1.3K	441.9 ± 1	2023-12-16	392K	450	`*.twitter.com,`
							`*.bbcrussian.com`

`KZ` [13]	2019-07-01 (+10)†	18.6 ± 19	81.5 ± 66	2019-07‡	37	138	`mail.ru, twitter.com`

TABLE 2: **Simulating censorship environments using three blocklists:** The "Top–10K" column shows the number of domains from the Tranco Top–10K that would be blocked based on the rules. Dynamic environments show the mean ± std across dates. †Non-consecutive dates. ‡We simulate a specific targeted event in Kazakhstan in 2019.

HTTPS interception in Kazakhstan in 2019 [13] (referred to in this work as `KZ`). We select these three blocklists as they are both significant and diverse: `CN` and `RU` represent blocking policies for two regions with extremely sophisticated and extensive blocking systems, with 222.2K and 392K rules, respectively. On the other hand, `KZ` represents a specific censorship event where 37 domains were targeted in a Man-in-the-Middle interception attack [13]. Furthermore, these blocklists also provide longitudinal data, that allows us to measure changes over time. For the relatively stable `CN` and `RU` blocklists, we select a 30-day period in 2023. The `KZ` blocklist represents data from 10 dates around a period in 2019 when the HTTPS interception was active, capturing the blocking being put into place and later removed. Our selection provides three environments that change differently from each other, enabling us to evaluate how well CenRL can capture diverse censorship patterns. Later in §5.2, we describe our real-world measurements in 15 other regions.

Controlled Static Environment (For *Task 1*). We use the three blocklists to create and simulate our static censorship environment. First, to implement the environment, we rely on BraveBlock [72], a Python adblock parser that takes in rules and matches them with HTTP requests, determining whether the requests are blocked. Note that the blocklists contain blocked content in the form of rules (e.g., *.example.com). Thus, for each blocklist, we transform each rule into a compatible format and feed it into BraveBlock. When CenRL selects a website to test, e.g., a.example.com, we turn it into an HTTP request and determine blocking using BraveBlock's `check_network_urls` function.

Controlled Dynamic Environment (For *Tasks 1 and 2*): We extend static environments to *dynamic* ones to represent how censorship naturally changes over time. For all three environments, we collect a set of consecutive dates with their corresponding blocklists. Then, for each blocklist, we create an instance of the environment (BraveBlock) per date that contains all the corresponding blocklist rules. To use a dynamic environment to simulate a run of CenRL, we first select a threshold, $w = 2,000$ that converts time steps to the number of censorship measurements per day. We choose this threshold to be analogous to Censored Planet's 2,000 measurements per measurement scan. Using this threshold with CenRL, we start simulating the dynamic environment with the BraveBlock rules from first date. Every website selected to be tested will be simulated using the environment corresponding to this date. Once we reach time step 2,000, we change the environment to the BraveBlock instance corresponding to the next available date. We repeat this until there are no more dates or time horizon T is reached.

Algorithm 1 CenRL Algorithm: Text in [ ]$_{ce}$ represents steps only for our controlled experiments and text in [ ]$_{re}$ represents steps only for our real-world experiments.

```
Require:
    Inputs:     List of $N$ Websites to test ($I_N$)
                Action Space ($\mathcal{A}$) and its Features ($F$)
                Reward Function ($R$) $\leftarrow$ Eq.(1) or (2)
                Policy ($\pi$) $\leftarrow$ UCB, $\epsilon$-greedy, Thompson Sampling
                [Blocklists over time ($S_L$): $|S_L| = 1$ in static env]$_{ce}$.
                (Static) Total Measurements ($w = 10K$)
                (Dynamic) Measurements per day ($w = 2K$)
                Time Horizon ($T =$[$w \cdot |S_L|$]$_{ce}$ | [$w\cdot$(# of days)]$_{re}$)
                [Training from previous measurements H (optional)]$_{re}$
    Output:  Set of blocked domains ($\mathcal{I_B}$)
procedure CenRL($I_N$, $F$, $T$, $w$)
    $\mathcal{A} \leftarrow$ BuildActionSpace($I_N$, $F$, [H]$_{re}$)
    [$S_{env} \leftarrow$ BuildEnv($S_L$)]$_{ce}$
    Env$\leftarrow$ [Next($S_{env}$)]$_{ce}$ | [CenRL API]$_{re}$
    $\mathcal{I_B} \leftarrow \emptyset$, $\mathcal{A}' \leftarrow \emptyset$
    for $t = 1$ to $T$ do
        $a_t \leftarrow$ ChooseArm($\pi$, $\mathcal{A}$)
        $i_t \leftarrow$ Random($a_t$)
        $B \leftarrow$ Env.Test($i_t$)
        $r_t \leftarrow R(a_t, i_t, B)$
        $\mathcal{A} \leftarrow$ Update($r_t, \pi, \mathcal{A}$)
        $a_t \leftarrow a_t \setminus \{i_t\}$
        $\mathcal{A}' \leftarrow \mathcal{A}' \cup i_t$
        if $a_t$ is $\emptyset$ then $\mathcal{A} \leftarrow \mathcal{A} \setminus \{a_t\}$, $\mathcal{A}' \leftarrow \mathcal{A}' \cup a_t$
        if $t$ mod $w$ == 0 then
            $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{A}'$, $\mathcal{A}' \leftarrow \emptyset$
            [Env$\leftarrow$ Next($S_{env}$)]$_{ce}$
        if $B$ is Blocked then $\mathcal{I_B} \leftarrow \mathcal{I_B} \cup i_t$
    return $\mathcal{I_B}$
```

4.3. CenRL Algorithm

Alg. 1 summarizes how CenRL operates as an end-to-end pipeline when given a set of N websites for testing and an initial action space ($\mathcal{A}$). In the controlled experiments (ref. §5.1), we also provide a list of blocklists corresponding to multiple dates (in the static case, only one blocklist date is used) to simulate our environment using the BuildEnv function. First, CenRL builds the action space from the set of websites and the feature used to construct arms, following

§4.1. For example, if the feature is rank bins, then the BUILDACTIONSPACE function creates one arm for each rank bin and populates it with the list of websites whose rank is within that bin. In our end-to-end pipeline for real-world measurements (ref. §5.2), we train and tune CenRL based on historical data from Censored Planet, and provide the tuning parameters as additional input to BUILDACTIONSPACE, which both helps select the best-performing feature for building arms, as well as set the initial expected reward values for each arm.

The algorithm initiates the environment using BUILDENV and NEXT, which is an iterator that loops through the list of dynamic controlled environments ordered by date, following §4.2. In real-world experiments, we use CenRL's API to access the network measurement environment. The CenRL algorithm runs from $t = 1$ to a time horizon T by selecting an arm a_t following a policy π, then a website i_t at random within the grouping of a_t. We note that instead of randomly selecting websites within arms, we can use other strategies, such as selecting websites in rank order within arms. Our investigation reveals that alternative strategies do not significantly alter performance. The algorithm then performs a censorship measurement for i_t, and returns a boolean for whether it is blocked, B, using our approach in §4.2. It uses B to calculate the reward, i.e., using Eq.(1) or Eq.(2) based on the task. It then updates our estimates of expected rewards, e.g., $Q_t(a)$, following the approach outlined in §3.4. Next, the algorithm removes the domain i_t from the arm and puts the arm to sleep if it no longer contains any domains. Once a day passes, i.e., $t \bmod w == 0$, it re-adds all removed domains and arms to the action space $\mathcal{A}$ and makes them awake again. This is because, for the next date, we expect censorship to change, such that certain domains may either be newly censored or no longer censored. We then iterate to the next environment in the case of controlled experiments, using this environment for the subsequent w time steps. Note that the expected reward is not reset across dates, as we expect CenRL to learn information over multiple dates.

5. Experiments

We describe the controlled and real-world experiments that we perform for evaluating CenRL. We use the three simulated environments described in §4.2 to conduct our controlled experiments, and perform real-world experiments using remote measurements to vantage points in 15 countries. In both experiments, we use the Tranco Top–10K list of popular websites as our input [12]. The Tranco list is the most common input test list used in measurement research and is also adopted by censorship platforms [4], [7]. We download the Top–10K list of domains (including subdomains) on December 21, 2023, and extract features such as the website category and TLD, as described in §4.1. We showcase an evaluation of the Tranco Top–100K list downloaded on the same date using the CN controlled environment in Appendix B, to show CenRL's performance on larger input sizes.

We also present an additional evaluation of the performance of manually-curated regional test lists in our three controlled environments. For this investigation, we used the Citizen Lab Regional Test Lists [11] for China, Russia, and Kazakhstan to define the action space of CenRL for the corresponding controlled environment. Each list is chosen from the date closest to the blocklist snapshot used to construct the corresponding static environment. All three lists are relatively small with 538, 827, and 167 domains for China, Russia, and Kazakhstan, respectively. This reflects their manual, community-driven curation efforts. The evaluation is showcased in §6.1.5.

5.1. Controlled Experiments

We conduct two sets of experiments with our controlled environments. First, for our static environments in §6.1.1, we conduct 10,000 measurements (T) for each action space feature (§4.1) i.e., we exhaust the entire action space. This experiment uses reward function R_1 and helps evaluate CenRL's performance in *Task 1*. Second, for dynamic environments in §6.1.2, we use a threshold (w) of 2,000 measurements per date to conduct measurements over 30 days for the `RU` and `CN` environments, and 10 dates for `KZ` (Table 2). We conduct dynamic environment experiments with both reward functions R_1 and R_2. We conduct a grid search to find the best-performing hyperparameters. In the rest of this section, we only report the best-performing hyperparameters, and a more detailed evaluation of hyperparameters is shown in Appendix D. We average results across 20 episodes for all controlled experiments.

5.1.1. Baselines. In our controlled experiments, we evaluate the performance of CenRL against a comprehensive set of baselines that are representative of approaches adopted by prior work and those that improve upon them using a strategic selection process that does not use machine learning. The baselines use the same Tranco Top–10K list, time horizon (T), and 20 episodes.

Tranco Random: At each time step, we randomly sample websites from the Tranco Top–10K, which is the approach adopted commonly by measurement platforms [4], [7].

Tranco Rank Order: We select websites from the Tranco Top–10K by rank order, an approach that has also been adopted by some previous work [13]. This baseline selects popular websites first, so we expect it to perform well in regions where popular websites are blocked (e.g., in our `KZ` environment). The *Tranco Random* and *Tranco Rank Order* baselines most closely resemble previous work.

Categories: This baseline improves on prior work by using a partially informed selection strategy. Websites are grouped by category and then a category is randomly chosen, and all sites within it are tested. Once a category is exhausted, a new category is selected.

Categories Round Robin: At each time step, we randomly select a category and a website within that category.

FOTN*	Country	AS	VP Subnet	Best Feature	c
Free	USA (US)	AS7018	12.47.31.x	Category	0.3
	Germany (DE)	AS3320	87.190.253.x	Category	0.03
	Costa Rica (CR)	AS11830	201.191.214.x	Subdomain	0.03
	Czechia (CZ)	AS30764	193.165.79.x	Category	0.03
	France (FR)	AS5410	89.91.71.x	Category	0.03
Partly Free	Pakistan (PK)	AS17557	221.120.226.x	Category	0.03
	Singapore (SG)	AS4773	203.127.53.x	Category	0.3
	Bangladesh (BG)	AS24432	202.134.12.x	Category	0.03
	Sri Lanka (SL)	AS18001	122.255.12.x	Subdomain	10
	Ecuador (EC)	AS27947	186.3.59.x	Category	0.03
Not Free	Turkmenistan (TM)	AS20661	95.85.97.x	Category	0.03
	UAE (AE)	AS15802	94.206.76.x	Category	0.03
	Cambodia (KH)	AS38235	203.176.143.x	Subdomain	1
	Myanmar (MM)	AS132148	103.70.249.x	Category	0.03
	Oman (OM)	AS28885	85.154.45.x	Category	0.03

TABLE 3: **Country and experiment selection for real-world evaluation:** Best-performing feature, exploration hyperparameter (c), and Autonomous System (AS) number and subnet for each vantage point. *FOTN=Freedom on the Net Status [1].

Categories Average Ranking: This baseline represents an approach where we consider both the category and average rank of websites within categories. First, the category arm with the highest average website ranking is selected, and then all websites within that arm are tested in a random order. Then, the category with the next highest average ranking is selected, and so on. We consider this baseline to be better informed than others, as it uses two features: website category and website rank.

Categories Average Ranking Round Robin: This baseline also first selects the category that has the highest average ranking, but then only selects one website within that category for testing. The baseline then moves to the category with the next highest average ranking and picks one website, proceeding in a round-robin fashion.

Entities, Entities Round Robin, Entities Average Ranking, and Entities Average Ranking Round Robin: These baselines work exactly like their category counterparts above, but with entities as the feature. We design these strategic baselines—extending beyond prior work–using the category and entity features as the corresponding CenRL variants perform well in static and dynamic environments respectively (§6).

5.2. Real-world Experiments

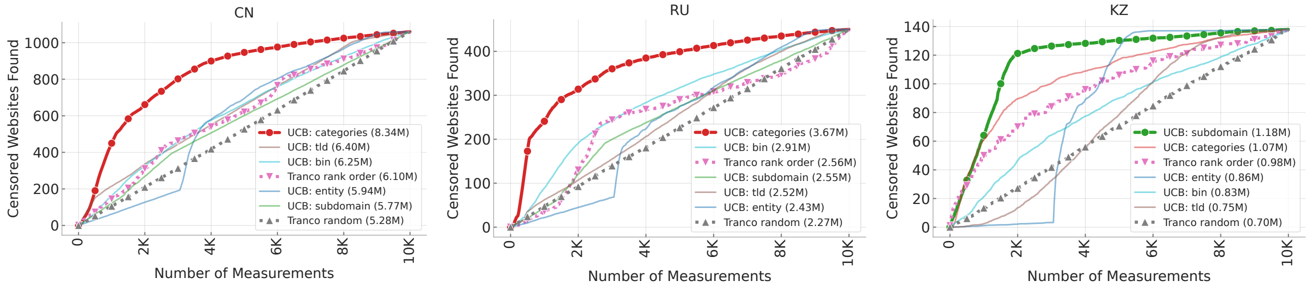
We evaluate the performance of CenRL using remote measurements to a randomly selected vantage point from Censored Planet's Hyperquack measurements observing censorship in top ASes in 15 countries, five of which are labeled as "Not Free" by the Freedom House Freedom on the Net report [1]—United Arab Emirates (AE), Cambodia (KH), Myanmar (MM), Oman (OM), Turkmenistan (TM) (We exclude `CN`, `RU`, and `KZ`)—five labeled as "Partially free"—Bangladesh (BG), Ecuador (EC), Pakistan (PK), Singapore (SG), Sri Lanka (SL)—and five labeled as "Free"—Costa Rica (CR), Czechia (CZ), Germany (DE), France (FR), United States (US). We select this distribution to showcase the generalizability of CenRL's policies to different censorship environments and rules. Table 3 provides a detailed list of the networks and subnetworks of our vantage points. Since we select one representative vantage point, our results do not generalize at the country-level. In addition, as we rely on Censored Planet's established remote measurement methodology, our approach might also detect blocking at the ISP and organization levels.

Real-world measurements with CenRL. Evaluating CenRL in a real-world setting for a specific region is non-trivial, as it requires hyperparameter tuning that involves thousands of real censorship measurements. Thus, to make CenRL practical, we build an end-to-end pipleline where we first tune its hyperparameters in a controlled environment using historical data from Censored Planet for each vantage point, utilizing the measurement efforts already underway in the community. For each vantage point, we download the latest data collected by Censored Planet (for 2,000 hand-curated and popular domains), and conduct hyperparameter tuning. We then use the top-performing hyperparameters and action space feature in our real-world experiments. Table 3 shows the best-performing action space feature and hyperparameters per vantage point from the Censored Planet data used for training. We observe that our policies generally encourage more exploitation (low c values). Note that the initial expected reward values are set per-arm after training with Censored Planet data. We emphasize that this best-effort approach at leveraging existing data opens up the opportunity for CenRL to be effectively applied to regions that are not known to have extensive blocklists. CenRL can also work with networks where existing data is not available, using its own measurements to learn about the environment.

Deploying CenRL. We integrate CenRL into Hyperquack [22], one of the tools used by Censored Planet to measure website blocking during the HTTPS request. We program Hyperquack to utilize CenRL's API to make use of its intelligent policies and inputs. For evaluation, we deploy both "CenRL Hyperquack" (which uses CenRL for inputs) and "vanilla Hyperquack" (with no modifications) for remote measurements, providing a point of comparison. We perform 2,000 HTTPS measurements per vantage point (similar to Censored Planet). For vanilla Hyperquack, we select the top 2,000 websites from the Tranco Top–10K, while we let CenRL's policy decide the selection from Tranco Top–10K for CenRL Hyperquack. Once a HTTPS measurement is made, we process the outcome according to Censored Planet's standard analysis pipeline which confirms blocking using specific signals [73], and provide the appropriate reward using R_1 (Eq.(1)).

5.3. Ethics Considerations

Censorship measurements require careful attention to the legal and technical risks directly or indirectly imposed on measurement hosts and endpoints. Fortunately, previous research, community discussions, and workshops on



(a) **CN (static)**

(b) **RU (static)**

(c) **KZ (static)**

Figure 2: **Evaluation of CenRL Static Environments vs. Tranco Baselines:** CenRL uses different features to accelerate the discovery of blocked websites. The parenthesis shows the area under the curve (AUC) for each line: the higher the value, the better the performance. The best-performing CenRL strategy and the two Tranco baselines are highlighted.

censorship detection have thoroughly explored these safety concerns, resulting in well-established technical practices that guide our work [7], [8], [10], [74], [75]. In building and evaluating CenRL, several of our decisions were based on the principle of minimizing risks and maximizing benefit. In fact, the core objective of CenRL is to minimize the number of measurements needed to study censorship more effectively, which not only reduces risk but also alleviates technical constraints on measurements networks and endpoints. Safety considerations also informed our key decision to conduct most large-scale evaluation in simulated environments, where extensive testing and hyperparameter tuning—spanning weeks and hundreds of thousands of measurements—was feasible without real-world risk and impact on measurement endpoints.

In addition to our controlled experiments, we perform real-world experiments to vantage points in fifteen countries, which involved establishing HTTPS connections with one remote web server vantage point in each country. These vantage points were carefully selected according to Censored Planet's established methodology, ensuring that they belong to large organizations (i.e., they are *infrastructural*), which are better equipped to handle our measurements compared to end-users. In addition, we follow good Internet citizenship by limiting our evaluation to 4,000 measurements per vantage point—the volume Censored Planet typically conducts over a week.. We separate measurements to the same vantage point by at least 10 seconds. We also follow all the best practices recommended by previous remote measurement research [7], [22], [27], [73], such as setting up WHOIS records, reverse DNS pointers, and web servers on our measurement client, all indicating that our measurements are part of a research project. We hope integrating our open-source CenRL into censorship measurement platforms will enable longitudinal evaluation in practice.

6. Results

In this section, we evaluate how CenRL performs in controlled (§6.1) and real-world (§6.2) censorship measurement environments.

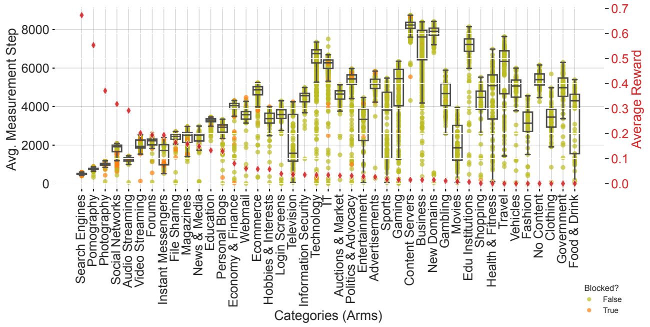
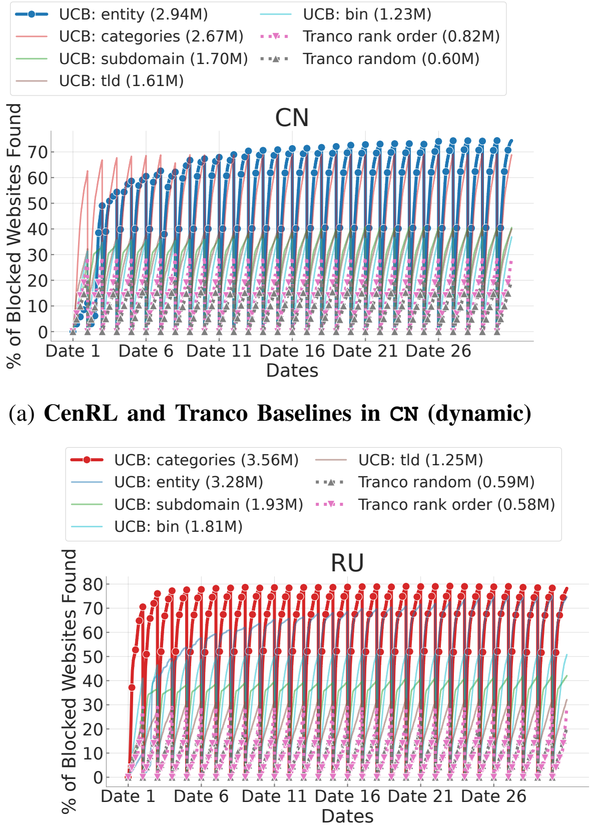


Figure 3: **Arm Selection Behavior for the Categories Feature in CN**: The box plot shows the category-wise distribution of the average time step at which a website is tested (left Y-axis). The red diamonds show the average reward obtained from testing the category (right Y-axis).

6.1. Controlled Evaluations

6.1.1. Static Environments. The results of our *Task 1* experiments with the static environment are shown in Fig. 2. Overall, we observe that CenRL significantly accelerates the discovery of blocked domains compared to the commonly used Tranco Random and Tranco Rank Order baselines. We compare CenRL to other baselines later in §6.1.3.

CN Static Environment. CenRL shows effective performance using the website categories feature in the CN environment (Fig. 2a). 89.3% of the blocked websites are discovered when half the measurements are done, representing a 79.6% increase in performance over the *Tranco Random* baseline and a 52.4% increase over the *Tranco Rank Order* baseline. CenRL discovers 75% of the blocked websites by measurement 2,944, as compared to measurement 7,509 for the *Tranco Random* baseline (155% faster). This shows that, when presented with a limited set of measurement resources (e.g., 2,000 measurements as in Censored Planet [7]), CenRL can find significantly more blocked websites, improving the utility of measurements. The TLD and rank bin features also perform well. The entity feature performs poorly initially due to the large number of potential



(a) **CenRL and Tranco Baselines in `CN` (dynamic)**

(b) **CenRL and Tranco Baselines in `RU` (dynamic)**

Figure 4: **Evaluation of CenRL in the `CN` and `RU` dynamic censorship environments:** In each date, we perform 2,000 measurements. We observe that CenRL learns more about the environment over time.

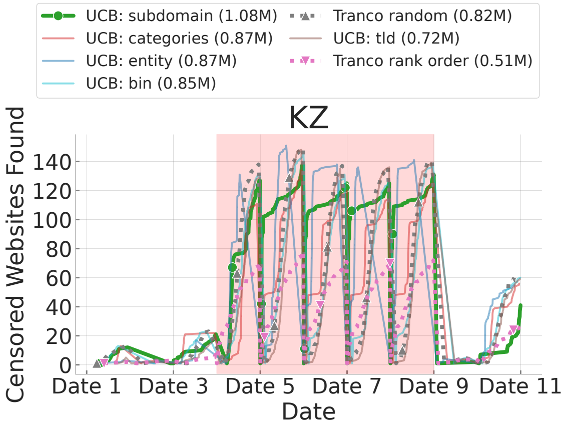


Figure 5: **Number of blocked websites found per day in the `KZ` dynamic environment**: The shaded red region shows the period of the censorship event. CenRL is able to learn the new blocking pattern quickly.

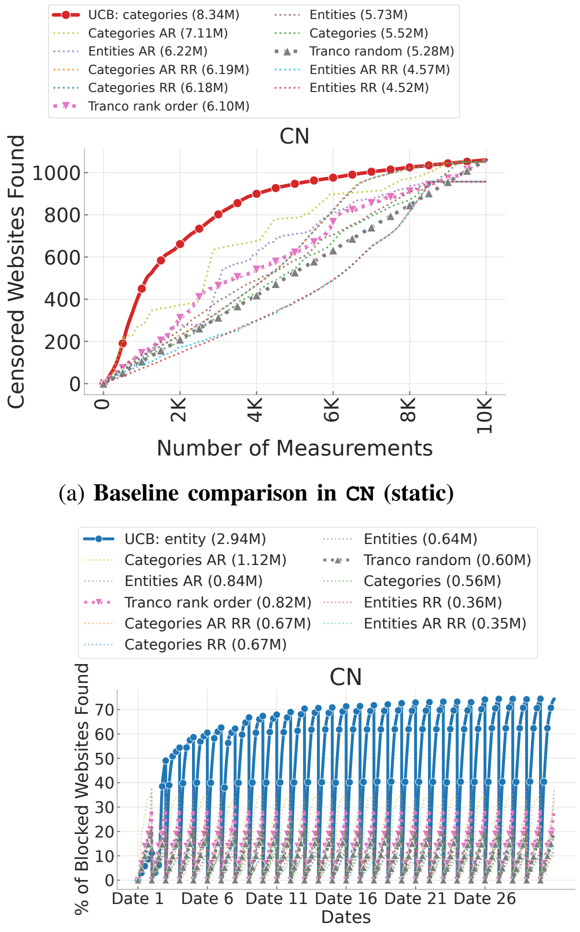
arms (ref. Table 1), but improves over time. As expected, the *Tranco Rank Order* baseline performs better than the *Tranco Random* baseline, as more popular domains are blocked in China. The strong performance of the category feature aligns with known censorship policies in China, which are often content-based. Fig. 3 shows when each category arm is tested and their average rewards, with high-reward arms (e.g., `Search Engines` and `Pornography`) tested earlier. Smaller arms with blocked websites, like `Audio Streaming`, are exhausted and put to sleep sooner, allowing later time steps to focus on learning from larger arms like `Social Networks`, enhancing overall efficiency.

`RU` Static Environment. The performance of CenRL in the `RU` static environment is similar to the `CN` case—the website categories feature performs the best, representing a 76.2% performance improvement over the *Tranco Random* baseline at the 5,000 measurements mark (Fig. 2b). CenRL finds 75% of blocked websites by measurement 2,486 while the *Tranco Random* baseline takes up to measurement 7,477 and *Tranco Rank Order* baseline takes up to measurement 7,486. Interestingly, we find that the rank bin and subdomain features perform better than they did in the `CN` case, indicating a difference in how censorship policies are implemented. CenRL reveals deeper insights regarding censorship policies in different regions and performs measurements accordingly.

`KZ` Static Environment. `KZ` represents a different environment compared to the two previous cases, where certain popular websites are targeted exclusively. As shown in Fig. 2c, the subdomain feature performs the best, possibly because major websites with common subdomains are blocked. For example, `m.facebook.com`, `m.youtube.com`, `m.vk.com`, and `m.ok.ru`, are all blocked. CenRL succeeds in finding more than 87.6% of blocked websites by the 2,000th measurement. The TLD feature performs the poorest; this is because most blocked websites have the '.com' TLD, which is the biggest arm.

6.1.2. Evaluation: Dynamic Environment. The results of our experiments in the dynamic controlled environments for *Tasks 1 and 2* are shown in Fig. 4 and Fig. 5.

`CN` and `RU` Dynamic Environments. These two environments reflect fairly stable blocklists over time. We find several interesting observations (ref. Fig. 4): (1) CenRL progressively discovers a higher portion of blocked websites over multiple days, showing that the agent has the ability to automatically learn blocking patterns over a period of time, even if the blocklist changes. This shows CenRL's practical applicability in longitudinal censorship measurement platforms such as Censored Planet and GFWatch. CenRL's learning plateaus in both of these environments (for example, around Day 5 in the `RU` case) showing CenRL's ability to learn censorship patterns within a few days. (2) CenRL strategies significantly outperform the baselines in *Task 1*, finding at the best case 148.17% more blocked websites than the *Tranco Rank Order* baseline and 263.16% more blocked websites than the *Tranco Random* baseline in the `CN` dynamic environment. In the `RU` case, CenRL finds 283.28% more blocked websites than the *Tranco Random* baseline in



(a) **Baseline comparison in `CN` (static)**

(b) **Baseline comparison in `CN` (dynamic)**

Figure 6: **Comparison of additional baselines in the static and dynamic `CN` environments.** AR–Average Ranking; RR–Round Robin

the best case. Moreover, CenRL's strategies can adapt to patterns over time, while baseline performance only depends on the blocklist. (3) While the categories and subdomain features perform well in the dynamic environment, the entity feature also performs surprisingly well, especially over a long period of time. Despite the large number of arms, we observe that CenRL is able to learn entity patterns over the larger number of measurements performed.

`KZ` Dynamic Environment. The `KZ` dynamic environment showcases a censorship event happening between date 4 and date 9 where 138 websites are temporarily blocked, as shown in Fig. 5. On the first date of the event, CenRL's strategy using the subdomain feature finds the first 100 blocked websites by measurement step 1,443 on average, while on the second and third dates, CenRL finds the first 100 blocked websites by measurement steps 158 and 173 respectively, showing that the agent can adapt quickly (*Task 2*). Since most of the blocked websites (Google, Meta, and X domains) belong to similar categories (`Social Networks`) and are highly ranked, the baselines perform considerably well too, although they are slow to identify the blocked websites. For instance, on the second day of the event, the *Tranco Random* baseline finds the 100th blocked website by measurement 1,075 i.e., CenRL can find the first 100 blocked websites about seven times faster than the state of the art. This shows CenRL's ability to perform well in *Task 2*, automatically responding to censorship events with no manual intervention.

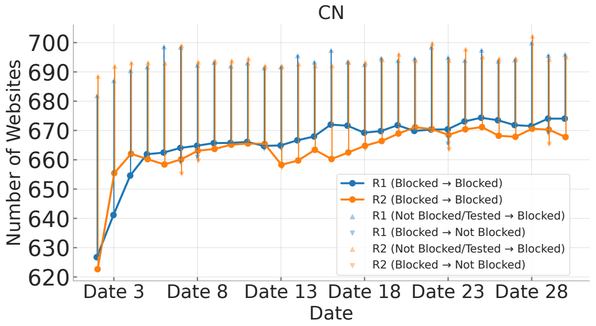
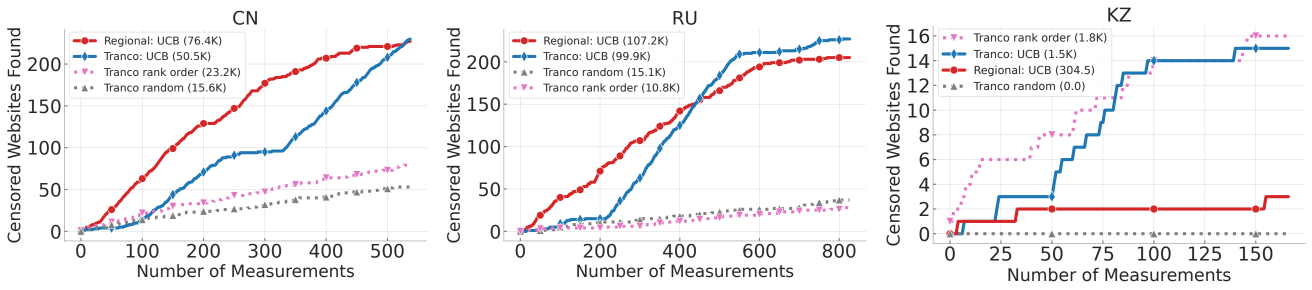


Figure 7: **Number of blocked websites found by reward functions R_1 and R_2 in `CN` with the categories feature**: The lineplot shows the number of blocked websites found on each date that were also found on the previous date, while ↑ shows the number of new blocked websites found. The ↓ shows the number of websites that changed from a blocked to not blocked state.

6.1.3. Results using other baselines. Figure 6 shows the results from evaluating the additional baselines in the `CN` static and dynamic environments (The insights for the `RU` and `KZ` environments are similar and are shown in Appendix E). We observe that our CenRL policies still significantly outperform all the additional "informed" baselines in both environments. We observe that the more-informed baselines relying on the average ranking perform better than the Tranco random and Tranco rank order baselines. This reinforces our finding that blocking policies follow certain patterns, where popular websites belonging to certain categories and entities are blocked together. Overall, we find that our intelligent RL policies are able to learn blocking patterns better and faster than other informed selection methods.

6.1.4. Evaluation: Reward Functions. Next, we explore the performance of reward function R_2 (§3.3) in our dynamic controlled environments. Recall that R_2 is geared towards *Task 2*, prioritizing finding new instances of blocking (Not Blocked → Blocked), and also rewarding cases where blocking is removed (Blocked → Not Blocked). Fig. 7 shows the number of blocked websites as well as changes in blocking found on each date using rewards R_1 and R_2 in the `CN` environment (categories feature). The performance of the rewards matches the different use-cases: R_2 finds new blocked websites quickly, as evidenced by the longer ↑ length in the early dates of the measurement. Moreover, R_2 also succeeds in finding the removal of blocking, as



(a) **CN (static)**

(b) **RU (static)**

(c) **KZ (static)**

Figure 8: **Evaluation of regional test lists from the Citizen Lab Test Lists in the CenRL Static Environments:** All UCB experiments are based on best performing hyperparameters from Figure 2. The parenthesis value shows the area under the curve (AUC) for each line. We observe that regional blocklists provide an early boost in performance in both CN and RU, but do not perform well in the new censorship event in KZ.

shown by the longer ↓ lengths. In contrast, R_1 is more consistent in finding the same blocked websites across dates. R_2 also finds changes in blocking more quickly on a specific date. For example, on the starting date of the Kazakhstan event (Date 4), R_2 (using subdomains) finds the first 100 blocked websites in measurement step 1,372 on average, 5.17% faster than R_1. Our evaluation shows that tailoring reward functions to use-cases can enhance findings.

6.1.5. Evaluation of Regional Test Lists. Figure 8 compares the performance of CenRL UCB when operating over the Citizen Lab regional test lists against the Tranco list. In the CN and RU environments, the regional lists provide a substantial advantage in early measurements. Since they are hand-curated, these lists include domains that the community already knows to be likely blocked, allowing CenRL to identify them quickly. By contrast, when using Tranco, a larger list with more uncertain domains, CenRL naturally requires more measurements to learn blocking patterns, but eventually catches up as the number of measurements increases. In contrast, the regional test list performs poorly in the KZ environment. By diving deeper to understand why, we find that many of the domains affected by the major interception event in Kazakhstan were absent from this regional list at the time, highlighting a key limitation of such lists: they are slow to respond to evolving censorship practice. In other words, the domain must be part of the test list for CenRL (or any other similar censorship measurement platform) to be tested and learned from. These results illustrate the trade-off for the community: use curated (small) regional lists that can provide valuable early performance boosts but are limited in scope and can miss certain new blocked domains vs. popularity-based lists that have a broader scale and coverage but require more measurements. Fortunately, for CenRL, there is a potential to combine both types of test lists to construct our action space.

6.2. Real-World Evaluations

We next explore CenRL's performance in real-world remote measurements, as described in §5.2. The results of our real-world experiment are shown in Fig. 9. CenRL Hyperquack identifies significantly more blocked websites in almost every vantage point compared to vanilla Hyperquack, which operates on a static list of 2,000 websites (top figure). The improvement is particularly pronounced in vantage points of "Not Free" countries; for instance, in the United Arab Emirates vantage point, CenRL Hyperquack detects 568 blocked websites, delivering a 5.6x increase over the 102 found by vanilla Hyperquack. Similarly, in vantage points of "Partly Free" countries, like Sri Lanka, CenRL Hyperquack identifies 499 blocked websites compared to vanilla Hyperquack's 215, achieving a 2.3x improvement. On average, across all vantage points, CenRL Hyperquack uncovers 2.75x more blocked websites. Exceptions include

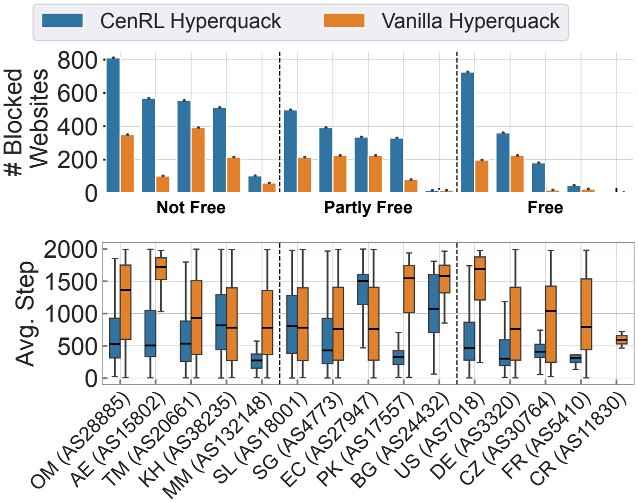


Figure 9: **Real-world experiments with vanilla Hyperquack and CenRL Hyperquack:** the top figure shows the overall number of blocked websites found, and the bottom figure shows the distribution of measurement time steps where blocked domains are found. We show the AS and country of the remote vantage point.

Bangladesh, where both methods detect 16 blocked websites; and Costa Rica, where vanilla Hyperquack finds 2 blocked websites when CenRL Hyperquack finds none. In such cases, the low level of blocking poses a challenge for the RL agent to optimize effectively. This is a limitation of any machine-learning model that lacks sufficient data to learn patterns. Performing a larger number of measurements or using regional test lists [11] may improve CenRL's performance in these cases.

Analyzing the websites identified by CenRL Hyperquack but not by the unmodified version, we find that CenRL is able to uncover distinct patterns that remain hidden otherwise. For example, in the vantage points in "Not Free" and "Partly Free" countries, CenRL was able to identify significant new blocking of websites falling under the `Social Networks` category (e.g., `internalfb.com` in MM vantage point, `mastodon.social` in OM vantage point, and `mixi.jp`) in SG vantage point. Other categories like `Pornography` and `Gambling` were also selected frequently, especially in "Partly Free" and "Free" countries (e.g., PK, US). We also find websites with subdomains `m` and `api` being blocked in measurements to the vantage point in SL, such as `m.vk.com` and `api.twitter.com`, showing that the blocking rules likely use regular expressions.

Figure 9 (bottom) also shows the measurement step at which blocked websites were detected across the two deployments. In addition to finding more blocked websites, we find that CenRL also finds blocked websites *quickly*. In vantage points in nine out of the 15 countries, CenRL Hyperquack identifies half of the total blocked websites it detects faster than when vanilla Hyperquack achieves the same milestone. For instance, in the Bangladesh vantage point, where both deployments find the same 16 blocked websites, CenRL Hyperquack identifies the first 8 by timestep 1001, whereas vanilla Hyperquack takes until measurement 1566. This demonstrates that even with fewer measurements, CenRL utilizes them efficiently by smartly selecting websites to uncover a significant portion of blocked websites.

7. Discussion & Conclusion

Extensions to CenRL. In this paper, we apply CenRL to discover blocked websites efficiently. The same policies can be applied to other aspects of censorship measurements, such as selecting measurement vantage points or protocols. For example, CenRL can use properties of vantage points like their AS number, network type, and geographic position to construct the action space. Moreover, while we explore the integration of CenRL into Hyperquack, CenRL's strategies can also be applied to other censorship measurement techniques, like Satellite/Iris [24], Augur [32], GFWatch [4], GFWeb [30], OONI [6], and ICLab [29]. In practical scenarios, performing multiple measurements in parallel is useful in enhancing speed, making future work on batch update policies for CenRL particularly valuable.

RL has applications in measurement problems other than censorship. For instance, RL can significantly enhance the detection of Internet shutdowns by selecting the right targets and measurement strategies on different protocols [31]. Moreover, Internet scanning (e.g., for vulnerabilities) can also benefit from intelligent decisions. Notably, CenRL's reward function can be adapted based on community needs, for example, to find unusual or important blocking. As the Internet landscape changes over time, CenRL's hyperparameters may need to be retrained and tuned. Users can rely on CenRL controlled environments (and build their own) to periodically (e.g., weekly) retrain for optimal performance.

Attacks on CenRL. CenRL can be targeted for attacks, for instance, by censoring authorities. The most apparent attack is to prevent CenRL from conducting censorship measurements necessary to calculate a reward, perhaps by dropping traffic from the measurement hosts. This challenge is not unique to CenRL but affects all censorship measurements. Although certain defenses can be employed (e.g., conducting control measurements from multiple hosts), measurement platforms cannot fully mitigate every attack. Fortunately, CenRL has the added benefit of being agnostic of measurement implementations, and can utilize an ensemble of approaches. Another attack can involve fingerprinting measurements performed by CenRL compared to normal web traffic. To combat this, we can employ techniques to obfuscate measurement patterns. Adversaries can also control the reward given to CenRL after each action. In response, we can update CenRL with more complex MAB algorithms designed to be robust against reward manipulation [76], [77]. However, there is no precedent of such attacks thwarting specific measurements, although the websites and probing software of censorship measurement platforms have been blocked previously [78].

Concluding Remarks. In this paper, we show the benefits of applying RL in censorship measurement. Our framework, CenRL, significantly outperforms the state of the art measurement process, highlighting that censorship policies tend to follow specific patterns that can be captured through RL. Additionally, RL's continuous learning ability can adapt to the evolving censorship landscape making it a more effective and practical option than traditional supervised and unsupervised learning methods. Overall, we find that incorporating intelligence into measurements significantly reduces the required time, resources, and manual efforts. We hope our work leads to a deeper understanding of censorship events threatening Internet freedom.

8. Acknowledgment

The authors thank the shepherd and reviewers for their comments which helped improve this paper. We acknowledge the censorship measurement community for providing open-source datasets that were central to this work. We also thank Nadav Oren for his contributions to the early stages of this project. This work was supported by the National Science Foundation under Grant Number CNS-2237552 and a Comcast Innovation Fund.

References

[1] Freedom House, “Freedom on the Net,” https://freedomhouse.org/report/freedom-net, 2024.

[2] R. Ramesh, R. Sundara Raman, M. Bernhard, V. Ongkowijaya, L. Evdokimov, A. Edmundson, S. Sprecher, M. Ikram, and R. Ensafi, “Decentralized Control: A Case Study of Russia,” in *Network and Distributed Systems Security Symposium (NDSS)*, 2020.

[3] M. Wu, J. Sippe, D. Sivakumar, J. Burg, P. Anderson, X. Wang, K. Bock, A. Houmansadr, D. Levin, and E. Wustrow, “How the Great Firewall of China Detects and Blocks Fully Encrypted Traffic,” in *USENIX Security Symposium*, 2023.

[4] N. P. Hoang, A. A. Niaki, J. Dalek, J. Knockel, P. Lin, B. Marczak, M. Crete-Nishihata, P. Gill, and M. Polychronakis, “How great is the Great Firewall? Measuring China’s DNS censorship,” in *USENIX Security Symposium*, 2021.

[5] S. Aryan, H. Aryan, and J. A. Halderman, “Internet censorship in Iran: A first look,” in *Free and Open Communications on the Internet (FOCI)*. USENIX, 2013.

[6] OONI, “Open Observatory of Network Interference,” 2012, https://ooni.org/.

[7] R. Sundara Raman, P. Shenoy, K. Kohls, and R. Ensafi, “Censored Planet: An Internet-Wide, Longitudinal Censorship Observatory,” in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2020.

[8] “NS Ethics ’15: Proceedings of the 2015 ACM SIGCOMM Workshop on Ethics in Networked Systems Research,” 2015.

[9] J. R. Crandall, M. Crete-Nishihata, and J. Knockel, “Forgive us our syns: Technical and ethical considerations for measuring internet filtering.” in *NS Ethics@ SIGCOMM*, 2015.

[10] B. Jones, R. Ensafi, N. Feamster, V. Paxson, and N. Weaver, “Ethical concerns for censorship measurement,” in *Ethics in Networked Systems Research*. ACM, 2015.

[11] Citizen Lab, “Censorship test list,” https://github.com/citizenlab/test-lists.

[12] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen, “Tranco: A research-oriented top sites ranking hardened against manipulation,” in *Network and Distributed Systems Security Symposium (NDSS)*, 2019.

[13] R. Sundara Raman, L. Evdokimov, E. Wustrow, A. Halderman, and R. Ensafi, “Investigating Large Scale HTTPS Interception in Kazakhstan,” in *ACM Internet Measurement Conference (IMC)*, 2020.

[14] S. Basso, M. Xynou, A. Filasto, and A. Meng, “Iran blocks social media, app stores and encrypted DNS amid Mahsa Amini protests,” https://ooni.org/post/2022-iran-blocks-social-media-mahsa-amini-protests/, 2022.

[15] R. Sundara Raman, L.-H. Merino, K. Bock, M. Fayed, D. Levin, N. Sullivan, and L. Valenta, “Global, Passive Detection of Connection Tampering,” in *ACM SIGCOMM*, 2023.

[16] A. Sarabi and M. Liu, “Characterizing the internet host population using deep learning: A universal and lightweight numerical embedding,” in *Internet Measurement Conference (IMC)*, 2018.

[17] R. Sommer and V. Paxson, “Outside the closed world: On using machine learning for network intrusion detection,” in *IEEE symposium on Security and Privacy (S&P)*, 2010.

[18] M. Gouel, K. Vermeulen, M. Mouchet, J. P. Rohrer, O. Fourmaux, and T. Friedman, “Zeph & iris map the internet: A resilient reinforcement learning approach to distributed ip route tracing,” *ACM SIGCOMM Computer Communication Review (CCR)*, 2022.

[19] B. Hou, Z. Cai, K. Wu, J. Su, and Y. Xiong, “6hit: A reinforcement learning-based approach to target generation for internet-wide ipv6 scanning,” in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021.

[20] G. Williams, M. Erdemir, A. Hsu, S. Bhat, A. Bhaskar, F. Li, and P. Pearce, “6sense: Internet-Wide IPv6 scanning and its security applications,” in *USENIX Security Symposium*, 2024.

[21] H. Le, J. Hayes, S. Elmalaki, M. Nasr, A. Markopoulou, M. Jagielski, Z. Shafiq, V. Sehwag, X. Guo, F. Tramèr *et al.*, “AutoFR: Automated Filter Rule Generation for Adblocking,” in *USENIX Security Symposium*, 2023.

[22] R. Sundara Raman, A. Stoll, J. Dalek, R. Ramesh, W. Scott, and R. Ensafi, “Measuring the deployment of network censorship filters at global scale,” in *Network and Distributed System Security Symposium (NDSS)*, 2020.

[23] Anonymous, “Towards a comprehensive picture of the Great Firewall’s DNS censorship,” in *Free and Open Communications on the Internet (FOCI)*. USENIX, 2014.

[24] P. Pearce, B. Jones, F. Li, R. Ensafi, N. Feamster, N. Weaver, and V. Paxson, “Global measurement of DNS censorship,” in *USENIX Security Symposium*, 2017.

[25] E. Tsai, D. Kumar, R. S. Raman, G. Li, Y. Eiger, and R. Ensafi, “CERTainty: Detecting DNS Manipulation using TLS Certificates,” in *Privacy Enhancing Technologies Symposium (PETS)*, 2023.

[26] S. Afroz and D. Fifield, “Timeline of Tor censorship,” 2007, http://www1.icsi.berkeley.edu/~sadia/tor_timeline.pdf.

[27] B. VanderSloot, A. McDonald, W. Scott, J. A. Halderman, and R. Ensafi, “Quack: Scalable remote measurement of application-layer censorship,” in *USENIX Security Symposium*, 2018.

[28] R. Sundara Raman, M. Wang, J. Dalek, J. Mayer, and R. Ensafi, “Network measurement methods for locating and examining censorship devices,” in *ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2022.

[29] A. Akhavan Niaki, S. Cho, Z. Weinberg, N. P. Hoang, A. Razaghpanah, N. Christin, and P. Gill, “ICLab: A Global, Longitudinal Internet Censorship Measurement Platform,” in *IEEE Symposium on Security and Privacy (S&P)*, 2020.

[30] N. P. Hoang, J. Dalek, M. Crete-Nishihata, N. Christin, V. Yegneswaran, M. Polychronakis, and N. Feamster, “GFWeb: Measuring the Great Firewall’s Web censorship at scale,” in *USENIX Security Symposium*, 2024.

[31] IODA, “Internet Outage Detection and Analysis,” https://ioda.inetintel.cc.gatech.edu/.

[32] P. Pearce, R. Ensafi, F. Li, N. Feamster, and V. Paxson, “Augur: Internet-wide detection of connectivity disruptions,” in *IEEE Symposium on Security and Privacy (S&P)*, 2017.

[33] Censored Planet, “Censored Planet Raw Data,” 2024, https://data.censoredplanet.org/raw.

[34] GFWatch, “GFWatch Dashboard,” https://gfwatch.org/, 2022, (Accessed on 08/30/2024).

[35] S. Nourin, E. Rye, K. Bock, N. P. Hoang, and D. Levin, “Is nobody there? good! globally measuring connection tampering without responsive endhosts,” in *IEEE Symposium on Security and Privacy (SP)*, 2025.

[36] J. Tang, L. Alvarez, A. Brar, N. P. Hoang, and N. Christin, “Automatic Generation of Web Censorship Probe Lists,” in *Privacy Enhancing Technologies Symposium)PETS)*, 2024.

[37] R. Ramesh, R. Sundara Raman, A. Virkud, A. Dirksen, A. Huremagic, D. Fifield, D. Rodenburg, R. Hynes, D. Madory, and R. Ensafi, “Network Responses to Russia’s Invasion of Ukraine in 2022: A Cautionary Tale for Internet Freedom,” in *USENIX Security Symposium*, 2023.

[38] M. Xynou, F. Arturo, M. Tawanda, and M. Natasha, “Zimbabwe protests: Social media blocking and internet blackouts,” 2019, https://ooni.org/post/zimbabwe-protests-social-media-blocking-2019/.

[39] B. Taye, “Sri Lanka: shutting down social media to fight rumors hurts victims,” 2019, https://www.accessnow.org/sri-lanka-shutting-down-social-media-to-fight-rumors-hurts-victims/.

[40] A. Aqil, K. Khalil, A. O. Atya, E. E. Papalexakis, S. V. Krishnamurthy, T. Jaeger, K. Ramakrishnan, P. Yu, and A. Swami, “Jaal: Towards network intrusion detection at ISP scale,” in *Conference on Emerging Networking EXperiments and Technologies (CoNEXT)*, 2017.

[41] D. Liu, Y. Zhao, H. Xu, Y. Sun, D. Pei, J. Luo, X. Jing, and M. Feng, “Opprentice: Towards practical and automatic anomaly

detection through machine learning," in *ACM Internet Measurement Conference (IMC)*, 2015.

[42] G. Marín, P. Casas, and G. Capdehourat, "Deepsec meets rawpower-deep learning for detection of network attacks using raw representations," *ACM SIGMETRICS Performance Evaluation Review*, 2019.

[43] G. Marin, P. Casas, and G. Capdehourat, "Rawpower: Deep learning based anomaly detection from raw network traffic measurements," in *ACM SIGCOMM Posters and Demos*, 2018.

[44] W. Li and A. W. Moore, "A machine learning approach for efficient traffic classification," in *International symposium on modeling, analysis, and simulation of computer and telecommunication systems*. IEEE, 2007.

[45] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE communications surveys & tutorials*, 2008.

[46] K. Bock, G. Hughey, X. Qiang, and D. Levin, "Geneva: Evolving censorship evasion strategies," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2019.

[47] S. Zhu, S. Li, Z. Wang, X. Chen, Z. Qian, S. V. Krishnamurthy, K. S. Chan, and A. Swami, "You do (not) belong here: detecting dpi evasion attacks with context learning," in *International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2020.

[48] P. Calle, L. Savitsky, A. N. Bhagoji, N. P. Hoang, and S. Cho, "Toward automated dns tampering detection using machine learning," *Free and Open Communications on the Internet (FOCI)*, 2024.

[49] J. Brown, X. Jiang, V. Tran, A. N. Bhagoji, N. P. Hoang, N. Feamster, P. Mittal, and V. Yegneswaran, "Augmenting rule-based DNS censorship detection at scale with machine learning," in *Knowledge Discovery And Data Mining*. ACM, 2023.

[50] E. Tsai, R. S. Raman, A. Prakash, and R. Ensafi, "Modeling and Detecting Internet Censorship Events," in *Network and Distributed Systems Security Symposium (NDSS)*, 2024.

[51] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time Analysis of the Multiarmed Bandit Problem," *Machine learning*, 2002.

[52] O. Avner and S. Mannor, "Concurrent bandits and cognitive radio networks," in *Machine Learning and Knowledge Discovery in Databases: European Conference*. Springer, 2014.

[53] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Transactions on Networking*, 2012.

[54] S. Maghsudi and E. Hossain, "Multi-armed bandits with application to 5g small cells," *IEEE Wireless Communications*, 2016.

[55] T. Yue, P. Wang, Y. Tang, E. Wang, B. Yu, K. Lu, and X. Zhou, "EcoFuzz: Adaptive Energy-Saving greybox fuzzing as a variant of the adversarial Multi-Armed bandit," in *USENIX Security Symposium*, 2020.

[56] J. Wang, C. Song, and H. Yin, "Reinforcement learning-based hierarchical seed scheduling for greybox fuzzing," in *Network and Distributed Systems Security Symposium (NDSS)*, 2021.

[57] Z. U. A. Tariq, E. Baccour, A. Erbad, M. Guizani, and M. Hamdi, "Network intrusion detection for smart infrastructure using multi-armed bandit based reinforcement learning in adversarial environment," in *International Conference on Cyber Warfare and Security (ICCWS)*. IEEE, 2022.

[58] F. Alharbi, M. Faloutsos, and N. Abu-Ghazaleh, "Opening digital borders cautiously yet decisively: Digital filtering in Saudi Arabia," in *Free and Open Communications on the Internet (FOCI)*. USENIX, 2020.

[59] A. Garivier and E. Moulines, "On upper-confidence bound policies for non-stationary bandit problems," *arXiv preprint arXiv:0805.3415*, 2008.

[60] Y. Wang, J.-Y. Audibert, and R. Munos, "Algorithms for infinitely many-armed bandits," *Advances in Neural Information Processing Systems*, 2008.

[61] R. Agrawal, "Sample mean based index policies by o(log n) regret for the multi-armed bandit problem," *Advances in applied probability*, 1995.

[62] Z. Fang, J. Wang, J. Geng, and X. Kan, "Feature selection for malware detection based on reinforcement learning," *IEEE Access*, 2019.

[63] V. Kuleshov and D. Precup, "Algorithms for multi-armed bandit problems," *arXiv preprint arXiv:1402.6028*, 2014.

[64] J. Vermorel and M. Mohri, "Multi-armed bandit algorithms and empirical evaluation," in *European conference on machine learning*. Springer, 2005.

[65] F. Trovo, S. Paladino, M. Restelli, and N. Gatti, "Sliding-window thompson sampling for non-stationary settings," *Journal of Artificial Intelligence Research*, 2020.

[66] D. J. Russo, B. Van Roy, A. Kazerouni, I. Osband, Z. Wen *et al.*, "A tutorial on thompson sampling," *Foundations and Trends® in Machine Learning*, 2018.

[67] Cloudflare, "Cloudflare API: Domain Intelligence," 2024, https://developers.cloudflare.com/api/operations/domain-intelligence-get-domain-details.

[68] K. Ruth, D. Kumar, B. Wang, L. Valenta, and Z. Durumeric, "Toppling top lists: Evaluating the accuracy of popular website lists," in *ACM Internet Measurement Conference (IMC)*, 2022.

[69] DuckDuckGo, "duckduckgo/tracker-radar: Data set of top third party web domains with rich metadata about them," https://github.com/duckduckgo/tracker-radar, Jul 2024, (Accessed on 07/25/2024).

[70] Disconnect Me, "Disconnect Tracking Protection," https://github.com/disconnectme/disconnect-tracking-protection, 2024.

[71] zapret info, "z-i," https://github.com/zapret-info/z-i, 2024.

[72] Intsights, "Braveblock: A fast and easy adblockplus parser and matcher based on adblock-rust package," https://github.com/Intsights/braveblock, (Accessed on 07/18/2024).

[73] R. Sundara Raman, A. Virkud, S. Laplante, V. Fortuna, and R. Ensafi, "Advancing the art of censorship data analysis," in *Free and Open Communications on the Internet (FOCI)*, 2023.

[74] A. Narayanan and B. Zevenbergen, "No encore for Encore? Ethical questions for web-based censorship measurement," *Technology Science*, 2015.

[75] D. Dittrich and E. Kenneally, "The Menlo Report: Ethical principles guiding information and communication technology research," U.S. Department of Homeland Security, Tech. Rep., 2012.

[76] P. Auer and C.-K. Chiang, "An algorithm with nearly optimal pseudo-regret for both stochastic and adversarial bandits," in *Conference on Learning Theory*, 2016.

[77] S. Ito, T. Tsuchiya, and J. Honda, "Adversarially robust multi-armed bandit algorithm with variance-dependent regret bounds," in *Proceedings of Thirty Fifth Conference on Learning Theory*, 2022.

[78] S. Basso, M. Xynou, and A. Filastò, "China is blocking OONI," https://ooni.org/post/2023-china-blocks-ooni/, 2023.

[79] M. Xynou, "Building a smart url list system: Policy for url prioritization," https://ooni.org/post/ooni-smart-url-list-system/, 2020.

[80] OONI, "URL Priorities," https://test-lists.ooni.org/prioritization, 2024, (Accessed on 08/30/2024).

Appendix A. OONI's Website Selection

The Open Observatory of Network Interference (OONI) is a censorship measurement platform consisting of a global community of volunteers that measure Internet censorship through crowdsourcing measurements [6]. While volunteers are able to test any custom website, the OONI probe measures websites on the Citizen Lab Test List by default, which is a list of manually compiled public-interest websites [11]. The test list consist of a list of websites that are globally interesting (1,660 websites in August 2024), and a list

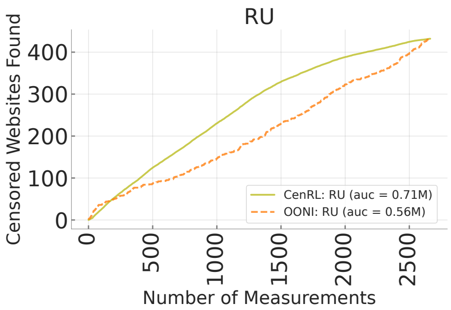


Figure 10: **Performance of OONI's URL prioritization and CenRL (categories feature) in `RU`:** We test 2,664 websites from the Citizen Lab test list.

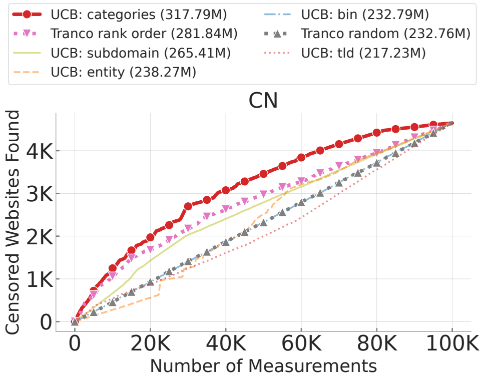


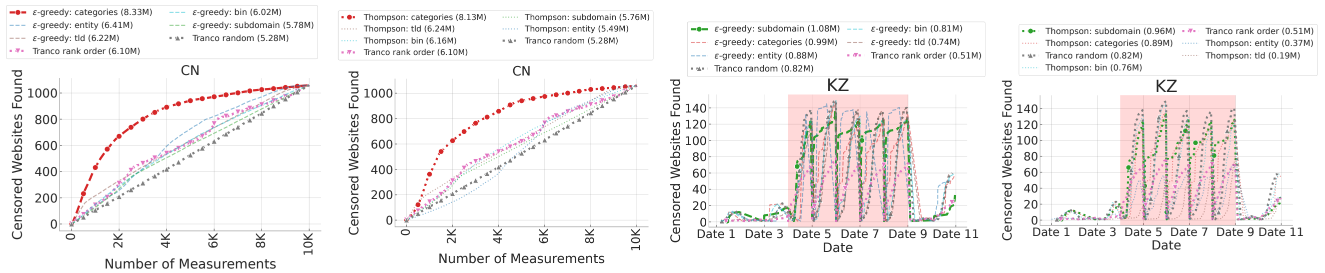
Figure 11: **Performance of CenRL with the larger Tranco Top–100K test list.**

specifically for each country (e.g., The Russia list has 1,092 websites). Since OONI probe runs are typically only 90 seconds long, and tests may take a variable amount of time, only some websites in the list can be tested. Therefore, to prioritize testing certain websites of public interest, the OONI project employs a URL prioritization system [79]. The process involves assigning a manual prioritization value to each category or website in the test list, with higher priorities assigned to content with higher public interest. For example, the category `NEWS` is given a priority value of 100 globally, while `PORN` is given a priority value of 30 [80]. During runtime, the OONI probe calculates a weight for each website in the test list by dividing the priority value with the number of measurements conducted for that website in the country for the past 7 days (or '0.1' if no tests have been conducted). Thus, websites with higher community interest and lesser number of measurements will be prioritized so that more useful data can be collected.

OONI's method contrasts with the RL approach used by CenRL, as OONI's prioritization scheme does not use the outcomes from each measurement to inform future measurements, which is the key insight for CenRL. Additionally, OONI's prioritization depends on volunteers or researchers updating priorities when censorship events occur, leading to potential delays and dependencies. Furthermore, OONI's website selection process is uniform across all users within a country and does not adapt based on ongoing measurements; its dynamic aspect is influenced by the number of measurements gathered in the previous week. Due to these differences, a direct comparison between CenRL and OONI's selection method is not applicable. However, to show the benefits and complementary nature of both approaches, we apply OONI's prioritization approach and CenRL's approach in testing the Citizen Lab global and Russia test lists [11] with the `RU` simulated static environment constructed using the blocklist downloaded on Aug 30, 2024. We note that we do not perform any initial training of Q-values for CenRL, which operates without any prior knowledge. In contrast, we obtain the OONI URL prioritization weights of the domains through the OONI API, and we use the category code from the Citizen Lab Global and Russia regional Test Lists to construct a new action space for CenRL. Simulating both cases in the `RU` environment, we find several interesting takeaways (ref. Fig. 10): First, the OONI prioritization system does extremely well at finding blocking initially, as many popular websites with extremely high weights (e.g., `www.facebook.com`) are blocked in Russia. This highlights the value of crowdsourcing: when available, community input can be extremely useful for prioritization. In future work, such pre-existing knowledge can easily be added to CenRL, in the form of higher initial expected rewards (Q_0) for certain websites or arms. Over time, CenRL is able to learn patterns regarding the blocking, and find blocked websites faster. This shows the value of learning patterns regarding the blocking itself, in contrast to using just the number of measurements. Overall, we see that OONI and CenRL's approaches are complementary to each other, and there is significant value in integrating both approaches together.

Appendix B. Evaluation of Tranco 100K

To explore whether CenRL can identify blocked websites within a larger list of domains, we repeat the static environment (`CN`-case) experiment with the Tranco list of Top–100K domains (over five episodes). The results are shown in Fig. 11: We see that CenRL's performance generally scales well over the larger list of domains, consistently finding blocked websites earlier than the Tranco baselines. Specifically, the categories feature again shows the best ability to discover blocked websites, finding 110.8% more blocked websites at measurement 20,000 compared to the Tranco Random baseline. We find that CenRL can find 50% of all blocked websites by step 25,902, while the baseline takes up to step 49,972. The Tranco rank order baseline performs well as many blocked domains are globally popular. Moreover, we see that the entity feature does not perform well initially, but its performance improves over time.

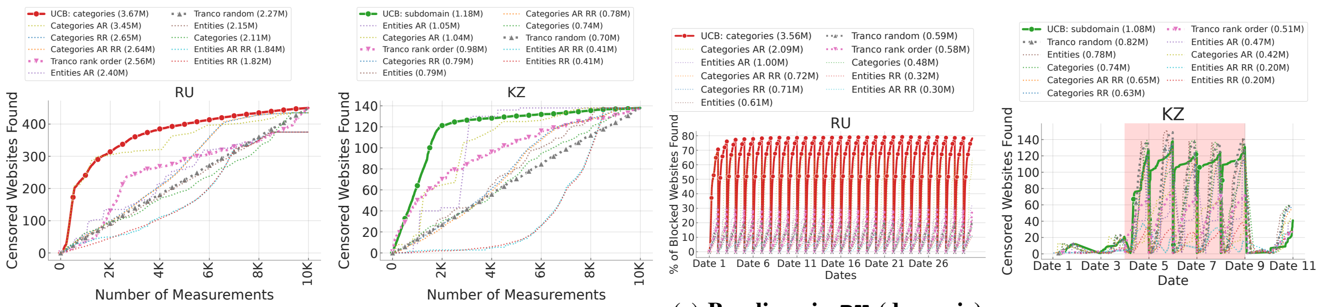


(a) **CN with ϵ-greedy (static)** (b) **CN with Thompson (static)** (c) **KZ with ϵ-greedy (dynamic)** (d) **KZ with Thompson (dyn)**

Figure 12: **Static environment results in CN and Dynamic environment results in KZ using ϵ-greedy sampling and Thompson sampling.**

Environment	Static (UCB)				Dynamic (UCB)			
	c	Step size	Init. value	auc	c	Step size	Init. value	auc
CHINA	0.3	0	0.6	8,342,397.45	0.3	0	0	654,793,677.07
	0.3	0	0	8,341,289.93	0.3	0	0.2	652,654,967.07
	0.3	0	0.8	8,339,354.72	0.03	0.2	0	574,681,408.55
	0.3	0	0.2	8,316,484.88	0.03	0	0	558,190,439.02
	0.3	0	0.4	8,313,870.35	0.03	0	0.2	554,653,551.00
RUSSIA	0.3	0	0	3,670,244.05	0.3	0	0	314,235,011.38
	0.3	0	0.6	3,665,866.75	0.3	0	0.2	313,267,203.92
	0.3	0	0.8	3,664,296.82	0.03	0	0	291,826,737.02
	0.3	0	0.2	3,661,731.50	0.03	0.2	0	291,742,056.30
	0.3	0	0.4	3,661,268.40	0.03	0	0.2	289,035,729.15
KAZAKHSTAN	0.03	0	0.2	1,065,909.25	0.3	0	0.2	5,112,710.50
	0.03	0	0.4	1,064,058.55	0.3	0	0	5,103,934.12
	0.03	0	0.8	1,059,741.30	0.03	0	0.2	4,712,758.28
	0.03	0.2	0	1,049,469.88	0.03	0.2	0	4,479,172.28
	0.3	0	0.6	1,049,352.65	0.03	0	0	4,445,937.97

TABLE 4: **Hyperparameter Characterization:** The top five best-performing hyperparameter combinations for CenRL's UCB policy for static & dynamic environments using the categories feature. 'c' controls the exploration vs. exploitation, the 'Step size' controls the learning rate (zero means 1 over the number of times the selected arm has been pulled), and 'Init. value' controls the initial expected reward. The 'auc' indicates the area under the curve in Fig. 2 and Fig. 4.



(a) **Baselines in RU (static)** (b) **Baselines in KZ (static)** (c) **Baselines in RU (dynamic)** (d) **Baselines in KZ (dynamic)**

Figure 13: **Baseline comparison in static and dynamic RU and KZ environments.** The top UCB result is plotted for comparison. The best-performing CenRL strategy and the two Tranco baselines are highlighted.

Appendix C. Other RL Policies

ϵ-greedy Sampling. The ϵ-greedy sampling policy presents a straightforward and effective strategy to address the explore-exploit dilemma [63], [64]. It defines a hyperparameter $\epsilon \in [0, 1]$ that controls the amount of exploration. At each time step, with probability ϵ, the policy selects an arm randomly (explore); otherwise, it selects the arm with the highest expected reward (exploit).

Thompson Sampling. Thompson Sampling is a Bayesian approach that models each arm with a beta distribution between [0,1], defined by two shape parameters, α and β. Initially, each arm has equal likelihood of being censored or not ($\alpha = 1$, $\beta = 1$). The distribution behaves as follows: (1) if $\alpha > \beta$, it skews towards 1; (2) if $\alpha \leq \beta$, it skews towards 0; (3) if $\alpha = \beta > 1$, it centers at 0.5. Thus, the beta-distribution represents the probability that an arm a is likely to have censored content. These shape parameters are updated as rewards are earned: $\alpha \mathrel{+}= r_t$ and $\beta \mathrel{+}= 1 - r_t$. The arm with the highest $Beta(S_\alpha, S_\beta)$ value is selected.

Appendix D. Hyperparameter Evaluation

In our evaluation, we use a grid search for hyperparameter tuning. The UCB and ϵ-greedy policies have 3 hyperparameters each, controlling the exploration vs exploitation ('c' $\in$ [0.03, 0.3,1,3] and '$\epsilon' \in [0.2, 0.4, 0.6, 0.8]$) (Thompson sampling controls exploration using the beta distribution), learning rate ('Step size' $\in [0, 0.2, 0.4, 0.8, 0.9]$, a zero Step Size represents 1 over the number of times the selected arm has been pulled), and the initial Q-value ('Init. value' $\in [0, 0.2, 0.4, 0.8, 0.9]$) set to all arms at the beginning of each episode. Table 4 shows the top five best-performing hyperparameter combinations in UCB when using the categories feature. Overall, we observe that the hyperparameter set showing the best performance encourages more exploitation (i.e., low values of 'c') and learning (i.e., low values of 'Step size'), showing that blocked websites contain patterns that our policies can exploit for intelligent measurements.

Appendix E. Other Baselines in RU and KZ environments

We show results, in Fig. 13, comparing our additional baselines (other than the Tranco baselines) to the CenRL UCB experiments in the RU and KZ environments. As noted earlier, the insights are very similar to the CN environment described in §6.1.3. The UCB strategy still outperforms other informed baselines, showing the usefulness of RL. The Categories average ranking and Entities average ranking baselines perform better than the Tranco baselines, utilizing the patterns we already build into CenRL's action space.

Results using other RL Policies. We present additional results from select controlled static (CN) and dynamic (KZ) environment experiments (ref. §5.1) using both ϵ-greedy sampling and Thompson sampling, shown in Fig. 12. Overall, we observe that both policies outperform the Tranco baselines, similar to UCB. We observe that UCB and ϵ-greedy sampling outperform Thompson sampling in most cases, possibly because the beta distribution in Thompson sampling takes more measurements to converge.

Appendix F. Meta-Review

The following meta-review was prepared by the program committee for the 2026 IEEE Symposium on Security and Privacy (S&P) as part of the review process as detailed in the call for papers.

F.1. Summary

This paper presents the CenRL system, which applies reinforcement learning to choosing which sites to test for blocking in a censored regime. CenRL models probe selection as a standard multi-armed bandit problem, where each arm represents a group of possible sites to probe grouped in some relevant way (e.g., category, domain, AS). Standard bandit algorithms can then be applied to obtain good performance, which is quantified by counting the number of blocks discovered for a given number of probes.

F.2. Scientific Contributions

- Creates a New Tool to Enable Future Scienc
- Provides a Valuable Step Forward in an Established Field

F.3. Reasons for Acceptance

1) More efficient and effective measurement tool for censorship
2) Novel integration of reinforcement learning into censorship measurement

F.4. Noteworthy Concerns

1) CenRL underperforms Hyperquack in some countries with low levels of blocking and to OONI's manual prioritization with fewer than 200 probes.

Appendix G. Response to the Meta-Review

The authors note that CenRL's approach is complementary to the manually curated priority values used by the OONI platform and can achieve improved early performance when initialized with existing data. See Appendix A for more information. We believe there is substantial value in combining such community-driven efforts with the adaptive, automated decision-making capabilities offered by reinforcement learning.