

USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Towards Practical Few-shot Multi-tab Website Fingerprinting

Lin Liu and Ziling Wei, *National University of Defense Technology;* Zhuotao Liu and Xinhao Deng, *Tsinghua University;* Zixuan Dong and Shuhui Chen, *National University of Defense Technology*

https://www.usenix.org/conference/usenixsecurity26/presentation/liu-lin

This paper is included in the Proceedings of the 35th USENIX Security Symposium.

August 12–14, 2026 • Baltimore, MD, USA

ISBN 978-1-939133-58-8

Open access to the Proceedings of the 35th USENIX Security Symposium is sponsored by



جامعة الملك عبدالله للعلوم والتقنية

King Abdullah University of Science and Technology

ARTIFACT EVALUATED
USENIX
AVAILABLE

Towards Practical Few-shot Multi-tab Website Fingerprinting

Lin Liu[1] Ziling Wei[1]✉ Zhuotao Liu[2]✉ Xinhao Deng[2] Zixuan Dong[1] Shuhui Chen[1]
[1] *National University of Defense Technology* [2] *Tsinghua University*

Abstract

Website fingerprinting (WF) attacks can infer the visited websites to deanonymize Tor networks by analyzing encrypted traffic patterns. Recent few-shot WF methods reduce reliance on large-scale data collection, yet they predominantly formulate WF as a single-label classification task and rely on meta-learning episodes that assume disjoint label sets and stable embedding spaces. These assumptions break down in the realistic multi-tab browsing, where traffic from multiple websites interleaves within a single observation window and the number of concurrent tabs is unknown. Meanwhile, the label space grows exponentially as the monitored set expands, making existing multi-tab WF methods costly to update. To address these challenges, we propose MMF, a novel framework for few-shot multi-tab WF. MMF shifts the meta-learning objective from single-label classification to *support-guided presence detection*. In each episode, we pair a mixed multi-tab query trace with a small set of single-tab support traces for each monitored website, and generate class-specific features by feature reweighting to decide which websites are present. This detection-centric formulation yields well-defined few-shot tasks in multi-tab scenarios and enables MMF to detect previously unseen websites from limited traces. We evaluate MMF on established public datasets and a new real-world dataset collected under varied browsing conditions. Results demonstrate that MMF consistently outperforms state-of-the-art multi-tab WF attacks across all settings. Notably, in the 5-shot scenario, MMF achieves improvements of up to 300% in Novel Precision@k, highlighting its strong capability for few-shot detection in dynamically growing website sets.

1 Introduction

Anonymous communication systems seek to protect both message content and communication metadata from network surveillance. Tor, one of the most widely deployed privacy-enhancing technologies for web browsing [30], pursues this goal by routing traffic through multi-hop relays and applying layered (onion) encryption to reduce the linkability between users and their online activities [14].

However, a persistent line of research, known as *website fingerprinting* (WF), demonstrates that passive adversaries can still infer visited websites by analyzing observable side channels, such as sequences of incoming and outgoing packets, inter-packet timing, and burst-level statistics [5, 8, 20, 33, 45, 47, 48]. Recent deep learning-based WF attacks have achieved high accuracy even under network noise and against various defenses [1, 32, 36, 40].

A critical barrier to a practical WF deployment is the *unsustainable cost of data collection and model maintenance*. Unlike many traffic classification tasks, in which class semantics are relatively stable, websites are dynamic monitored targets whose traffic patterns drift due to content updates, browser changes, and network dynamics. To retain effectiveness, attackers must continuously recollect labeled traces at scale and retrain models, an operationally expensive process. These constraints have motivated *few-shot* (low-data) WF, which aims to preserve strong inference performance using only a handful of labeled traces per website [7, 28, 40, 50, 52, 59, 60].

However, most prior work on few-shot WF implicitly assumes a *single-tab* setting, where each trace corresponds to an isolated visit to one website. In practice, web browsing is often *multi-tab*: users open multiple tabs within a short time window, causing their traffic to interleave within a single observation window. This interleaving corrupts the clean, site-specific fingerprints exploited by classic WF attacks, while the unknown and dynamic tab count further complicates the problem [13, 49, 55]. Recent studies have begun to revisit multi-tab WF with end-to-end models and report promising performance under more realistic conditions

✉ Corresponding author.

[10, 13, 18, 22, 31, 49, 53, 55, 56, 58]. Nevertheless, the multi-tab setting is typically more data-intensive than single-tab WF because the combinatorial diversity of tab-opening patterns substantially expands the space of possible mixtures, increasing the labeled data required for adequate coverage and robust performance.

This discrepancy prompts a fundamental question: *Why has few-shot WF research remained confined to the single-tab setting, rather than extending effectively to the more realistic multi-tab setting?* We argue that the key obstacle is not simply "harder inputs", but a *shift in the learning objective* from *classification* to *detection*.

Existing few-shot WF methods [28, 40, 59, 60] for the single-tab setting typically follow either metric-based or optimization-based few-shot learning paradigms under a classification-centric formulation adapted from single-label classification [41,44]. These paradigms are effective because episodic training implicitly relies on three properties: (i) query traces and their supports are drawn from comparable distributions, (ii) each query corresponds to exactly one website class, and (iii) a globally coherent embedding geometry allows queries to be aligned with class prototypes or stable adaptation directions.

However, these properties no longer hold in multi-tab browsing. A query trace can contain multiple interleaved websites, so its representation is not expected to align with any single-class prototype [6, 25, 57]. As a result, prior few-shot WF formulations [7, 28, 40, 50, 52, 59, 60] can not directly extend to realistic multi-tab scenarios, leaving a critical gap between data-efficient methods and practical attack settings.

To bridge this gap, we present *MMF*, a framework for *few-shot multi-tab* website fingerprinting that makes episodic few-shot task construction feasible under the multi-tab setting via *support-guided feature reweighting*. Given a mixed multi-tab query trace, MMF encodes it into a feature map that preserves local burst-level microstructures, which remain informative even when multiple websites are interleaved. For each target website, MMF summarizes its K clean *single-tab* support traces into a class-conditioned reweighting vector. This vector performs channel-wise gating on the multi-tab query features to produce a class-specific feature map, and a *presence* head then predicts whether the target website appears in the mixed traces.

This detection-centric formulation, inspired by object-detection paradigms, keeps few-shot episodes well defined in multi-tab settings. Each meta-training episode pairs (i) one *multi-tab* query trace with (ii) a K-shot *single-tab* support set for each website in a sampled target set, and supervises learning with per-class binary presence labels. Within an episode, the learner is trained to derive class-conditioned reweighting vectors from supports, use them to amplify target-class evidence while suppressing interference in the mixed query representation, and optimize independent presence predictions. As a result, MMF learns transferable mechanisms for extracting class-specific weights from single-tab traces, rather than relying on a globally coherent prototype geometry or brittle few-step adaptation.

After base training, MMF enables *continual onboarding* of previously unseen websites using only a handful of newly collected clean single-tab traces and a small number of labeled multi-tab traces for lightweight fine-tuning, substantially reducing the need for expensive long-horizon collection and full retraining over large-scale mixed-traffic datasets.

The contributions of our work are given as follows.

- **Problem reformulation:** We identify the task mismatch that prevents conventional few-shot WF from scaling to multi-tab browsing, and reframe multi-tab WF as a few-shot presence detection problem.
- **Framework design:** We propose *MMF*, a novel framework which formulates few-shot multi-tab WF as *support-guided presence detection* and learns transferable class-conditioning via support-conditioned feature reweighting with episodic training, enabling fast adaptation to unseen websites with only limited data. By producing class-specific representations from a detection perspective, MMF further removes the need for any prior knowledge of the tab count using a *single* shared model.
- **Extensive evaluation:** We evaluate MMF on both public and self-collected datasets under closed-world, open-world, defense-aware, and dynamic tab settings. The results demonstrate that MMF significantly outperforms state-of-the-art baselines, achieving up to 300% higher Novel Precision@k in 5-shot scenarios while maintaining robust performance as the monitored set expands.

Our work bridges the gap between data efficiency and real-world applicability in website fingerprinting, advancing state-of-the-art towards practical deployment. We release code and data to support result reproducibility, as well as future exploration in this field.

2 Background

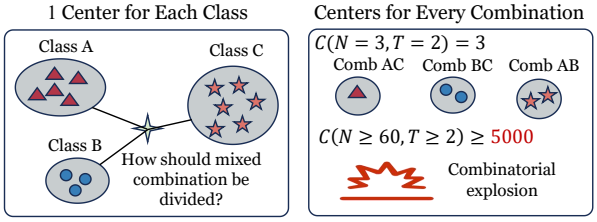
2.1 Website Fingerprinting Attacks

Website fingerprinting (WF) aims to infer which websites a user visits by passively analyzing traffic side channels (e.g., packet direction sequences, burst patterns), even when the application payload is encrypted and the intended destination is hidden by anonymity systems such as Tor [14, 33, 47].

Table 1: Summary of representative works.

Works	Multi-Tab[1]	Few-Shot[2]	Tabs-Unknown[3]	Detection[4]
DF [39]	×	×	×	×
BAPM [18]	✓	×	×	×
MMD [42]	×	✓	×	×
NetCLR [3]	×	✓	×	×
DQF [60]	×	✓	×	×
TMWF [22]	✓	×	×	✓
ARES [13]	✓	×	✓	×
FMWF [31]	✓	✓	×	×
MMF	✓	✓	✓	✓

[1] **Multi-Tab**: Users can open multiple tabs while browsing.
[2] **Few-Shot**: Using few samples to train the model for identifying unseen classes.
[3] **Tabs-Unknown**: The number of tabs that a user opens is dynamic and unknown.
[4] **Detection**: Uses an object-detection paradigm as the framework.



Note: N means number of classes, T means number of tabs opened by users

Figure 1: **The dilemma of FSL.** Left: Adopting one center per class fails because a mixed trace containing multiple websites cannot align with any single-class prototype. Right: Alternatively, defining a unique center for every possible combination leads to a combinatorial explosion in the number of required prototypes (e.g., $C(N,T)$), making the metric space enormous.

Early WF studies relied on hand-crafted features and conventional classifiers, and showed that encrypted traffic remains distinguishable [19, 32, 33, 47]. With the adoption of deep learning, WF attacks have increasingly shifted toward end-to-end representation learning, substantially improving accuracy and challenging WF defenses by extracting richer discriminative patterns directly from raw traces [4, 11, 12, 36, 39]. However, these high-performing models typically assume abundant and continuously refreshed training data. The adversary must repeatedly collect a large number of traces per monitored website, and retrain models whenever the monitored set or deployment environment changes [9, 49]. This maintenance burden has motivated a growing body of few-shot (low-data) WF research, including episodic meta-learning for rapid adaptation, metric- or contrastive-based objectives for learning discriminative embeddings under data scarcity, and augmentation expansion to mitigate label sparsity [3, 7, 29, 40, 52, 60]. Notably, existing few-shot WF works almost exclusively adopt a *single-tab* assumption, where each trace corresponds to browsing activity from a single tab, making the problem naturally amenable to episodic few-shot *classification*.

Recent work has therefore revisited multi-tab WF with more systematic end-to-end architectures, including Transformer-based designs and multi-tab-native frameworks [13, 18, 22, 27, 31, 53, 56, 58]. Most existing multi-tab WF methods formulate the task primarily as a multi-label classification problem, following the conventional WF perspective [33, 39], while TMWF [22] is among the few works that consider WF from an object-detection-inspired perspective. However, TMWF does not fully address the tab-unknown setting and lacks few-shot capabilities. From a maintenance-cost perspective, multi-tab WF calls for stronger few-shot capabilities than single-tab WF, because the space of feasible label combinations grows combinatorially with the monitored set size and the number of concurrently opened tabs. ARES [13] introduces a one-vs-all framework to support new-class onboarding, but it still requires non-trivial training samples and maintains a separate model for each monitored class, making the model-maintenance cost scale linearly with the monitored set size. FMWF [31] explores general few-shot strategies based on data augmentation and transfer learning, but it lacks effective designs tailored to the multi-tab scenario, leaving substantial room for improvement (see experimental results in § 5.3).

2.2 Few-Shot Learning

Few-shot learning (FSL) studies generalization when only a small number of labeled examples are available per class. A dominant paradigm is episodic (meta-task) training: each episode consists of a support set and a query set, and the learner is trained over many episodes to acquire an inductive bias for rapid generalization from limited support data [16, 34, 41, 44]. In prior WF work, few-shot attacks largely fall into two families. First, *metric-based* approaches learn an embedding space and classify query examples by their distance to class prototypes or nearest neighbors; WF instantiations often adopt Triplet/Quadruplet-style objectives to promote separability under data scarcity [29, 40, 60]. Second, *optimization-based meta-learning* targets rapid adaptation to novel classes, typically via a small number of gradient-based update steps on the support set (e.g., MAML-style methods) [7, 16]. Although effective for *single-tab* traces, these designs largely assume a *classification-centric* formulation of the task.

A key challenge is that extending these *classification-centric* few-shot paradigms to multi-tab WF is not merely a matter of increased difficulty, but a shift in the underlying task logic. Metric-based methods implicitly assume

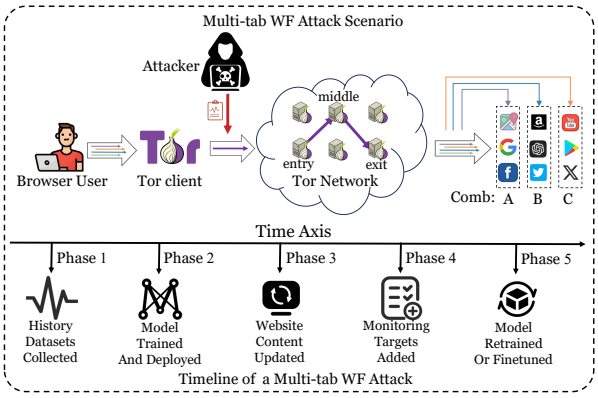


Figure 2: **Threat model.** The adversary passively monitors encrypted traffic between the user and the Tor entry node.The timeline illustrates the practical maintenance challenge: when content evolves and new targets are added (Phases 3-4), the attacker must efficiently update the model using limited newly collected data (Phase 5).

a stable global geometry that a query example should lie close to its class prototype in the embedding space. In the multi-tab scenario, temporal interleaving can break globally coherent single-tab fingerprints, causing a mixed trace to fall far from *all* single-tab prototypes. Alternatively, defining prototypes for each *combination* of co-occurring websites leads to a combinatorial explosion in the number of centers [2, 43] (we illustrate this collapse in Figure 1). Optimization-based meta-learning faces a related *task-construction dilemma*. When support traces (single-tab) and query traces (multi-tab) are drawn from markedly different distributions, a few adaptation steps may not bridge the gap. Moreover, updates can become unstable when multiple sources of evidence coexist within a single mixed trace.

3 Threat Model

We consider website fingerprinting (WF) attacks against Tor. The adversary is a *passive* network observer, and it cannot decrypt application-layer content and does not modify, inject, delay, or drop traffic. Instead, it records externally observable metadata on the client-side link (e.g., on the client-to-entry (guard) relay link, at a local gateway, or from an ISP vantage point), including packet directions and timestamps [14, 33, 47]. Following prior WF work, the adversary can also obtain labeled training data by repeatedly visiting monitored websites, and train/evaluate models [9, 36, 37, 39]. Figure 2 summarizes our threat model.

Beyond the conventional *single-tab* assumption, we focus on *multi-tab* browsing, where a user may open multiple tabs concurrently. Traffic streams from different websites then become temporally *interleaved* within the observation window, yielding a mixed multi-tab trace rather than an isolated single-website trace [13, 22, 53, 55, 56]. Another key aspect of our setting is that the number of concurrently opened tabs is *dynamic* and *unknown a priori* to the attacker at inference time, while most prior works choose to ignore this [18, 22, 27, 31, 53, 55].

As illustrated by the lower half of the timeline in Figure 2, we also consider *target dynamics* over longer time horizons. In practice, an attacker typically pre-trains and deploys a model using historical data. However, the monitored set may evolve (e.g., newly added targets), and website content and resource-loading behavior can drift over time. These changes introduce distribution shift that can quickly degrade the reliability of a trained model [10, 23, 49]. From a maintenance standpoint, we assume the attacker can periodically update the deployed model using limited newly collected data, including a small number of labeled single-tab and multi-tab traces. We use these data to fine-tune the deployed model, avoiding prolonged collection and full retraining on large-scale datasets.

Finally, following prior WF evaluations, we consider both *closed-world* (CW) and *open-world* (OW) settings. In CW, all visited websites belong to the monitored set. In OW, the user may visit a large and unknown set of unmonitored websites. These pages generate background traffic that can be interleaved with monitored traffic, and the attacker must detect whether monitored websites are present despite this realistic interference.

4 Design of MMF

4.1 Overview

As discussed in § 1, extending few-shot website fingerprinting (FS-WF) from single-tab traces to realistic multi-tab browsing is not simply a "harder-input" variant. Multi-tab traffic temporally interleaves encrypted segments from multiple websites. This violates the single-label assumption and disrupts trace-level patterns that conventional episodic few-shot *classification* implicitly depends on [41]. The problem is further complicated by the fact that the number of concurrently opened tabs is dynamic and unknown.

To make episodic few-shot task construction well defined under multi-tab mixing, MMF draws inspiration from feature reweighting in few-shot detection [25] and reformulates FS-WF as *presence prediction*. Rather than deciding *which* class a mixed trace belongs to, MMF determines, for each target website, *whether* evidence of that website is present in the mixed trace. Concretely, for a target class c, MMF summarizes its K single-tab supports into a class-conditioned reweighting vector, uses it

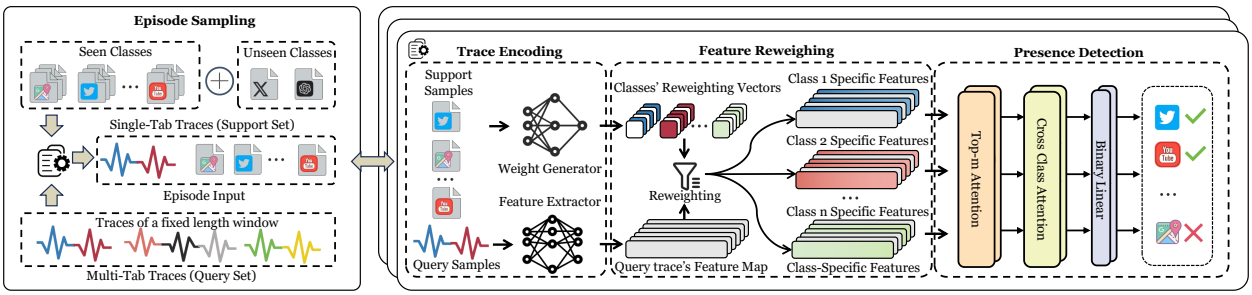


Figure 3: **MMF overview.** Each episode pairs a mixed multi-tab query trace with K clean single-tab support traces for each target website. MMF encodes the query into a feature map, derives a class-conditioned reweighting vector from the supports, reweights query channels to form a class-specific feature map, and predicts per-class presence probabilities.

to modulate the channel semantics of the multi-tab query feature map, and trains a presence head with per-class binary supervision. This reformulation avoids reliance on globally coherent prototype geometry and brittle few-step adaptation. It instead learns a transferable mechanism for extracting class-specific evidence from obfuscated mixtures.

Figure 3 gives an overview of the MMF framework. Training begins with an Episode Sampling module that constructs meta-learning tasks. Each episode samples K single-tab traces per target class as supports, pairs them with a multi-tab query trace, and uses multi-hot presence labels as supervision. Within each episode, MMF performs forward computation and parameter updates through three modules, namely Trace Encoding, Feature Reweighting, and Presence Detection.

Trace Encoding. MMF encodes a multi-tab query trace into a multi-channel feature map that preserves local, burst-level microstructure and avoids premature global pooling. These local patterns, induced by resource fetches, can remain informative even when multiple websites are interleaved. In parallel, the support branch encodes single-tab traces and produces a compact class-conditioned reweighting vector that captures which channel patterns are characteristic of the target class.

Feature Reweighting. Conditioned on this vector, MMF reweights the channels of the query feature map, amplifying target-related responses while suppressing interference from other websites and background traffic, to produce a more robust class-specific feature map.

Presence Detection. Finally, the detection head aggregates the class-specific feature map and outputs the probability that the target website appears in the mixed trace. To remain sensitive to sparse, scattered evidence while mitigating diffuse correlations induced by heavy mixing, MMF incorporates two attention mechanisms and optimizes a weighted binary objective to handle the extreme class imbalance.

4.2 Episode Sampling

The *Episode Sampling* module constructs training tasks for MMF in both *meta-training* (base) and *meta-fine-tuning* (few-shot). We follow a deployment-oriented workflow in which an attacker can typically collect labeled traces for a set of monitored *base/seen* websites over a long period, enabling offline meta-training. After deployment, the monitored set may expand or evolve. Newly added *novel/unseen* websites typically come with limited labeled traces, so the model must be updated without expensive full retraining.

During meta-training, we adopt episodic learning and reorganize the dataset into a large number of episodes. For both query and support traces, we use only the packet *direction sequence*. We encode outgoing packets as $+1$ and incoming packets as -1. Each multi-tab query trace is represented as $x_j^q \in \{-1,+1\}^{L_q}$ and each single-tab support trace as $x_{c,k}^s \in \{-1,+1\}^{L_s}$, where L_q and L_s denote fixed sequence lengths after truncation/padding.

We denote each episode as $\mathcal{T}_j = (\mathcal{S}_j, Q_j)$. Specifically, the query set contains one labeled multi-tab example, $Q_j = \{(x_j^q, \mathbf{y}_j^q)\}$ with $\mathbf{y}_j^q \in \{0,1\}^C$, and the support set provides K single-tab traces per class, $\mathcal{S}_j = \{x_{c,k}^s \mid c = 1,\ldots,C;\ k = 1,\ldots,K\}$. Here, C is the number of target classes sampled in the episode, and $\mathbf{y}_j^q$ is a multi-hot *presence* label indicating which of these C websites appear in the multi-tab query. Unlike conventional single-tab FS-WF episodes formulated as a mutually exclusive classification task, MMF episodes are driven by *presence supervision*. The learner must decide, for each target website, whether it appears in the same obfuscated multi-tab query trace.

Meta-fine-tuning follows the same episode construction but uses drastically less data for novel websites. For each unseen website, the attacker has limited labeled single-tab and multi-tab traces that contain the novel website for fine-tuning. Accordingly, fine-tuning episodes use few-shot budgets for both supports and queries. We also

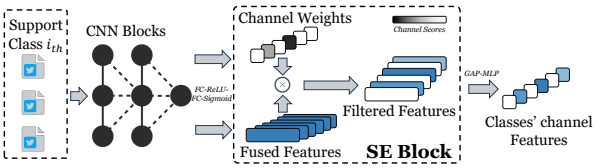


Figure 4: **Weight generator.** MMF summarizes K single-tab support traces of class c into a class-conditioned reweighting vector $\mathbf{W}_c$ by support encoding, shot aggregation, SE-style channel attention, and an MLP projection.

subsample seen classes to comparable few-shot sizes so that updates are not dominated by abundant base data, thereby focusing adaptation on onboarding and calibrating novel websites.

We emphasize that support traces are used only during training and fine-tuning. After training, MMF precomputes and stores the class-conditioned reweighting vector for each monitored website. At deployment time, the online detector takes only a multi-tab query trace as input and applies the stored class-conditioned vectors for class-aware processing and presence prediction, avoiding extra parameters.

4.3 Trace Encoding

MMF builds on an observation commonly exploited in deep-learning WF [13]: even when global trace-level patterns are disrupted, website-specific *local* loading behaviors (e.g., short bursts and direction alternations induced by resource fetches) can remain informative and be captured by convolutional encoders. Accordingly, MMF's trace encoder produces (i) a *query feature map* that preserves *where* local patterns occur and (ii) *class-conditioned reweighting vectors* that summarize *which* channel patterns characterize each target website. These outputs share the same number of channels, enabling class-aware channel filtering that is robust to position shifts in later stages.

Feature extractor. For each multi-tab query trace, MMF adopts a DF-style 1D convolutional encoder that has proven effective for WF representation learning [39]. The encoder maps the input direction sequence to an intermediate feature map whose sequence dimension preserves the *positions* of local patterns, while the channel dimension encodes *types* of discriminative local signatures. Formally, given a labeled multi-tab query trace x_j^q, the feature extractor outputs

$$\mathbf{F}_j \;=\; f_\theta(x_j^q) \in \mathbb{R}^{L' \times D}, \tag{1}$$

where L' is the reduced sequence length after convolution/pooling, and D is the channel dimension shared across classes. This "position response + channel semantics" representation is well suited to interleaving because the same website may contribute local patterns at varying positions.

Weight generator. Given a target website class c with K clean single-tab support traces $\{x_{c,k}^s\}_{k=1}^K$, MMF generates a class-conditioned reweighting vector $\mathbf{W}_c \in \mathbb{R}^D$ as a compact channel-wise descriptor of class c. As illustrated in Figure 4, each support trace is first encoded by a 1D CNN encoder into a channelized feature map:

$$\mathbf{U}_{c,k} \;=\; g_\phi(x_{c,k}^s) \in \mathbb{R}^{L' \times D}. \tag{2}$$

We then aggregate the K shot-level feature maps by averaging:

$$\mathbf{U}_c \;=\; \frac{1}{K}\sum_{k=1}^{K} \mathbf{U}_{c,k} \in \mathbb{R}^{L' \times D}. \tag{3}$$

To reduce sensitivity to shot-level noise when K is small, we apply an SE-style channel attention block that computes channel weights from $\mathbf{U}_c$:

$$\begin{aligned} \mathbf{a}_c &= \sigma\Big(\mathrm{MLP}_{\mathrm{SE}}\big(\mathrm{GAP}(\mathbf{U}_c)\big)\Big) \in \mathbb{R}^D, \\ \widetilde{\mathbf{U}}_c &= \mathbf{U}_c \odot \mathbf{a}_c, \end{aligned} \tag{4}$$

where $\mathrm{GAP}(\cdot)$ averages over the L' positions, $\sigma(\cdot)$ is the sigmoid function, and $\odot$ denotes channel-wise multiplication with broadcasting over positions. Finally, we obtain a vector summary by global average pooling and map it to $\mathbf{W}_c$ via an MLP:

$$\mathbf{v}_c = \mathrm{GAP}(\widetilde{\mathbf{U}}_c) \in \mathbb{R}^D, \quad \mathbf{W}_c = \mathrm{MLP}(\mathbf{v}_c) \in \mathbb{R}^D. \tag{5}$$

By construction, $\mathbf{W}_c$ lies in the same channel space as the query feature map in Eq. (1), which enables direct class-aware channel filtering in the next stage.

4.4 Feature Reweighting

§ 4.3 produces a query feature map $\mathbf{F}j \in \mathbb{R}^{L' \times D}$ and class-conditioned reweighting vectors $\mathbf{W}c_{c=1}^C$, where channels correspond to different local pattern types. In multi-tab traces, a target website may contribute patterns at arbitrary positions due to interleaving. MMF therefore does *not* attempt to learn a globally aligned template along the sequence dimension. Instead, inspired by [25], MMF uses $\mathbf{W}c$ to filter the *channel semantics* of $\mathbf{F}j$ in a position-agnostic manner. At each position, target-related channels are emphasized while irrelevant channels are attenuated, yielding a more stable class-specific representation under mixing and temporal shifts.

As shown in Figure 5, we implement feature reweighting as a 1D dynamic convolution whose kernel is parameterized by the reweighting vector $\mathbf{W}c$. Given $\mathbf{W}_c \in \mathbb{R}^D$

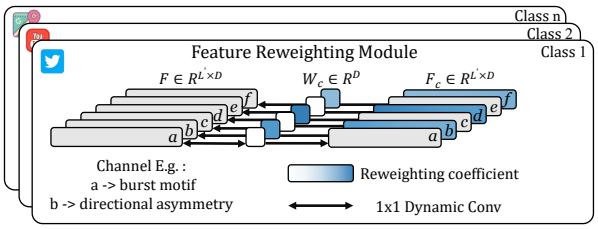


Figure 5: **Support-guided feature reweighting.** The support-derived vector $\mathbf{W}_c$ performs channel-wise gating on the query feature map $\mathbf{F}_j$ (equivalently, a dynamic depthwise 1×1 Conv1D), producing a class-specific feature map $\mathbf{F}_{j,c}$.

and $\mathbf{F}_j \in \mathbb{R}^{L' \times D}$, the reweighted feature map for class c is defined as

$$\mathbf{F}_{j,c} \;=\; \mathrm{DynConv1D}(\mathbf{F}_j; \mathbf{W}_c) \in \mathbb{R}^{L' \times D}. \tag{6}$$

In our implementation, this operation reduces to channel-wise gating at each position.

$$\begin{aligned} &\mathbf{F}_{j,c}[t,d] = \mathbf{F}_j[t,d] \cdot \mathbf{W}_c[d], \\ &t = 1,\ldots,L',\ d = 1,\ldots,D. \end{aligned} \tag{7}$$

We apply Eq. (6) to each target class and obtain $\{\mathbf{F}_{j,c}\}_{c=1}^{C}$, which are then fed into the presence detection head (§ 4.5).

Why channel reweighting is the right "matching" operation. The crucial difference between this stage and the SE block in § 4.3 is *where* the channel weights come from and *what* they represent. SE performs *intra-sample* refinement by using a trace's own global summary to modulate its channels. In contrast, MMF's reweighting is *cross-sample* and *class-conditioned*. The vector $\mathbf{W}c$ is derived from the single-tab supports of class c and specifies which channel semantics should be highlighted in the multi-tab query. This design yields a support-guided matching operation that is robust to interleaving and positional shifts, laying the foundation for per-class presence prediction.

4.5 Presence Detection

Given the class-conditioned feature maps $\{\mathbf{F}_{j,c}\}_{c=1}^{C}$ produced in § 4.4, MMF performs *presence detection* to determine whether each target website is present in an obfuscated multi-tab trace. Although feature reweighting suppresses class-irrelevant channels, heavy interleaving can still induce long-range and diffuse correlations in the sequence representation. We therefore adopt Top-m self-attention (as used in ARES [13]) to retain only the most salient interactions. Target evidence can be sparse and scattered, whereas interference tends to create diffuse yet uninformative correlations. Pruning attention to the top-m links mitigates this diffusion while preserving a small set of informative long-range dependencies.

In addition, real browsing sessions may exhibit biased co-occurrence among websites, leading to weak but consistent *cross-class* dependencies in multi-tab traces. Although we do not inject any explicit preference over website combinations during training, we incorporate a cross-class attention module to model contextual interactions across classes. This allows each class decision to benefit from other classes' representations in the realistic multi-tab traces, without requiring any prior knowledge or explicit enumeration of website combinations.

Next, we formulate presence detection. Let $\hat{\mathbf{F}}_{j,c} \in \mathbb{R}^{T \times D}$ denote the reweighted feature sequence for class c (we use $T = L'$). We first add positional embeddings to obtain

$$\mathbf{H}_{j,c} \;=\; \hat{\mathbf{F}}_{j,c} + \mathbf{P}, \quad \mathbf{P} \in \mathbb{R}^{T \times D}. \tag{8}$$

We then apply L=2 stacked Top-m self-attention layers:

$$\mathbf{H}_{j,c} \leftarrow \mathrm{TopMAttn}(\mathbf{H}_{j,c}), \quad \text{repeated } L \text{ times}, \tag{9}$$

where each layer computes scaled dot-product attention and masks all but the top-m attention scores per query position before softmax normalization. We aggregate along time to obtain a class vector:

$$\mathbf{g}_{j,c} \;=\; \mathrm{Pool}(\mathbf{H}_{j,c}) \in \mathbb{R}^{D}, \tag{10}$$

where $\mathrm{Pool}(\cdot)$ is implemented as mean pooling over the T positions.

To incorporate cross-class context, we stack the per-class vectors into $\mathbf{G}j = [\mathbf{g}j,1;\ldots;\mathbf{g}j,C] \in \mathbb{R}^{C \times D}$ and apply $L\times$=2 layers of cross-class attention:

$$\mathbf{G}_j \leftarrow \mathrm{CrossClassAttn}(\mathbf{G}_j), \quad \text{repeated } L_\times \text{ times}. \tag{11}$$

Finally, an MLP produces an independent presence logit for each class:

$$z_{j,c} = \mathrm{MLP}(\mathbf{G}_{j,c}), p_{j,c} = \sigma(z_{j,c}), c = 1,\ldots,C, \tag{12}$$

where $\sigma(\cdot)$ is the sigmoid function and $p_{j,c}$ is the predicted probability that website c is present in query j.

Training objective. Given the multi-hot presence label $\mathbf{y}_j \in \{0,1\}^C$ for query j, we optimize a weighted binary cross-entropy to address extreme class imbalance:

$$\begin{aligned} \mathcal{L}_{\mathrm{WBCE}} = \frac{1}{C}\sum_{c=1}^{C}\Big(&-\alpha y_{j,c}\log\sigma(z_{j,c}) \\ &-(1-y_{j,c})\log\big(1-\sigma(z_{j,c})\big)\Big), \end{aligned} \tag{13}$$

where $\alpha > 1$ up-weights positive classes and is set according to the empirical positive ratio. This objective keeps the detector sensitive to rare positives while preventing the optimization from being dominated by abundant negatives.

5 Evaluation

In this section, we evaluate MMF on the *few-shot multi-tab* WF task using both public datasets and a self-collected dataset. We compare MMF against representative state-of-the-art baselines under matched experimental settings. Our evaluation addresses the following questions: (1) **Base Performance.** Does MMF preserve strong *base* multi-tab performance, or does it trade off base performance for few-shot adaptability? (2) **Few-Shot Adaptability.** How does MMF perform in the standard few-shot multi-tab setting when onboarding unseen websites? (3) **Scalability.** How does the performance of MMF evolve as the number of unseen classes grows? (4) **Robustness to Tab Dynamics.** Can MMF remain robust under *tab-count mismatch* (dynamic and unknown numbers of tabs) during both base training and few-shot fine-tuning? (5) **Resistance to Defenses.** Is MMF more robust than prior works when different defense mechanisms are deployed? (6) **Open-World Robustness.** Is MMF's *presence-detection* formulation robust in the *open-world* setting with unmonitored background traffic? (7) **Stress Test.** Is MMF more robust than baselines in higher tab counts and heavier interleaving scenarios? (8) **Real-World Drift Effectiveness.** Is MMF more effective than baselines in multi-tab settings under more realistic network conditions? (9) **Analysis of Inference Overhead.** Is the framework of MMF more efficient than prior onboarding frameworks?

5.1 Experimental Setup

Implementation. We implemented MMF in PyTorch. The hyperparameters of MMF are listed in Table 2.

Datasets. We use two data sources: (i) a self-collected dataset with real multi-tab traces and aligned single-tab traces, and (ii) synthetically mixed multi-tab traces constructed from a widely used public single-tab WF dataset [12, 39].

Self-collected dataset. We extend the ARES data-collection pipeline [13] to enable simultaneous acquisition of (a) mixed multi-tab traces and (b) their corresponding single-tab traces on the same machine. Concretely, we first select 50 popular websites based on public popularity rankings. For each tab count setting in {3,4,5}, we randomly generate 20,000 website tab combinations and visit each of them three times. In parallel, we collect 30 single-tab visits per website to construct the single-tab traces. The collection spans approximately two months and uses servers located in Singapore. Further details are provided in Appendix A.2.

We further select 25 additional websites to expand the monitored set. For each newly added website, we generate only 25 combinations and collect 20 visits per combination, resulting in a data-limited scenario.

Table 2: Key hyperparameters used in our evaluation.

Component	Hyperparameter	Value
Episode sampling	Query length (L_q)	20000
	Support length (L_s)	10000
	Shots per class (K)	5
Input / dimension	Input channels	2 (direction, mask)
	Output channels (D)	256
Model	DF blocks	4
	#Top-m self-attn layers (L)	2
	#Cross-class attn layers ($L_\times$)	2
Training	Loss	Weighted BCE
	Dropout	0.15
	Optimizer	Adam
	Learning rate	5×10^{-5}
	Weight decay	10^{-4}

Public dataset with multi-tab synthesis. Existing public multi-tab datasets typically do not provide aligned single-tab visits for the same websites, which limits their use in support-guided few-shot evaluation. We therefore construct multi-tab traces for 60 base classes through controlled synthesis from public single-tab datasets. The resulting synthetic benchmark covers closed-world (CW) [39], open-world (OW) [39], and defended settings. For the defended setting, we use WTF-PAD [24] and Front [17], consistent with ARES [13]. All datasets are constructed from the same data source, WFlib [12], to ensure consistency across settings.

The synthesis code is built on TMWF's mixing tool [22], which reports that synthetic traces can be harder to classify than multi-tab traces collected in the wild. We further improve the mixer to capture more realistic temporal relationships. Specifically, the original implementation enforces a strict overlap-ratio constraint to avoid cases where a long trace fully subsumes a shorter one. We relax this constraint and allow both partial overlap and full containment between tabs, which better reflects real browsing. We present details of the algorithm and data splits in Appendix A.1. The defended synthetic benchmark is used in § 5.6, the open-world synthetic benchmark is used in § 5.7, and the self-collected dataset is used only in § 5.9. The remaining experiments are conducted on the closed-world synthetic benchmark. Unless otherwise specified, we set $r_{\max} = 0.4$ in the synthesis algorithm.

Note that, for few-shot adaptation, we synthesize only a small number of samples for 30 novel websites. Each synthesized trace contains one novel website mixed with several base websites, and each novel website appears exactly K times in the query pool to match the K-shot setting.

Baselines. We compare MMF with representative multi-tab WF baselines, including BAPM [18], FMWF [31],

TMWF [22], and ARES [13]. These methods cover mainstream design choices for multi-tab WF and few-shot multi-tab WF. We do not include few-shot *single-tab* WF methods as additional baselines because most assume mutually exclusive *classification* and rely on globally coherent metric geometry or fixed decision boundaries. These assumptions do not directly transfer to multi-tab traces [28, 40, 59, 60].

Metrics. Following the common multi-label evaluation protocol used in prior multi-tab WF studies [13], we report ROC-AUC and Precision@Top-k (P@k) for standard multi-tab evaluation. Let $p_{i,c} \in [0,1]$ denote the predicted presence score of class c for query i, and let $y_{i,c} \in \{0,1\}$ denote the corresponding ground-truth presence label. We compute macro-averaged ROC-AUC over all monitored classes:

$$\text{AUC} = \frac{1}{C}\sum_{c=1}^{C} \text{AUC}_{\text{ROC}}\left(\{y_{i,c}\}_{i=1}^{N}, \{p_{i,c}\}_{i=1}^{N}\right). \quad (14)$$

For P@k, we rank all monitored classes by their predicted scores for each query i. Let k_i denote the number of ground-truth monitored classes present in query i, and let Top-$k_i(p_i)$ be the corresponding set of top-ranked classes. We compute

$$\text{P@k} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{k_i}\sum_{c\in\text{Top-}k_i(p_i)} y_{i,c}. \quad (15)$$

For few-shot evaluations, we further consider whether newly incorporated websites can be identified from limited data. Accordingly, we report Novel P@k, which measures the precision of novel-class predictions selected in the Top-k_i set. Let $\mathcal{C}_{\text{novel}}$ denote the set of novel classes. We have

$$\text{Novel P@k} = \frac{\sum_{i=1}^{N}\sum_{c\in\mathcal{C}_{\text{novel}}}\mathbb{I}[c\in\text{Top-}k_i(p_i)]\cdot y_{i,c}}{\sum_{i=1}^{N}\sum_{c\in\mathcal{C}_{\text{novel}}}\mathbb{I}[c\in\text{Top-}k_i(p_i)]}. \quad (16)$$

Novel Recall@k (R@k) is a recall-form variant based on the same calculation principle of Novel P@k, which can be formulated as:

$$\text{Novel R@k} = \frac{\sum_{i=1}^{N}\sum_{c\in\mathcal{C}_{\text{novel}}}\mathbb{I}[c\in\text{Top-}k_i(p_i)]\cdot y_{i,c}}{\sum_{i=1}^{N}\sum_{c\in\mathcal{C}_{\text{novel}}} y_{i,c}}. \quad (17)$$

5.2 Closed-World Multi-tab Evaluation

Few-shot capability should not come at the expense of standard multi-tab performance. In practice, an attacker may still collect substantial training data before deployment and train a model offline. Maintaining competitive performance in this standard multi-tab setting is therefore a prerequisite for a credible few-shot multi-tab framework.

Table 3: Closed-world (CW) multi-tab performance on the base monitored set. We report macro ROC-AUC and P@k for 3/4/5-tab settings.

Method	3-tab		4-tab		5-tab	
	AUC	P@k	AUC	P@k	AUC	P@k
BAPM	0.9076	0.7100	0.8783	0.6835	0.8352	0.6745
FMWF	0.9825	0.9268	0.9698	0.8966	0.9490	0.8534
TMWF	0.9791	0.9193	0.9746	**0.9346**	0.9757	**0.9372**
ARES	0.9887	**0.9466**	0.9823	0.9260	0.9725	0.8962
MMF	**0.9952**	0.9313	**0.9920**	0.9256	**0.9899**	0.9215

We first evaluate MMF in the closed-world (CW) multi-tab setting and compare it with representative baselines. Table 3 reports AUC and P@k for 3/4/5-tab settings. MMF achieves the best AUC performance across all tab counts. In the 3-tab setting, MMF achieves an AUC of **0.9952**, outperforming the strongest baseline (ARES). Moreover, as the number of tabs increases from 3 to 5, MMF exhibits slight performance degradation, while other baselines suffer more noticeable drops. This stability suggests that the detection-centric formulation is less sensitive to tab counts. We further validate robustness under dynamic tab settings in § 5.5. Note that TMWF shows an increase in P@k, which is mainly due to its fixed maximum-tab setting, set to 6 by default [22].

5.3 Multi-tab Few-shot Adaptation on Unseen Websites

After establishing strong CW performance, we evaluate MMF in the core *few-shot multi-tab* setting. Starting from the checkpoint pre-trained in § 5.2, we introduce unseen websites as novel classes and perform few-shot fine-tuning using only K labeled traces per novel website. We vary $K \in \{5, 10, 15, 20, 30\}$ and evaluate under 3-5 tab queries, yielding 15 scenarios. Figure 6 reports **Novel P@k**.

The results show that existing multi-tab WF baselines exhibit limited few-shot adaptability. In extremely low-shot settings, baselines achieve low Novel P@k, and performance degrades further as the number of tabs increases from 3 to 5. This suggests that these approaches largely rely on decision structures learned in fully supervised CW training and adapt poorly with only a handful of new samples. We also tried both full-model fine-tuning and head-only updates with the hyperparameters from their open-source code, but neither produced satisfactory gains; BAPM can learn novel classes but exhibits severe base-class forgetting.

In contrast, MMF consistently outperforms all baselines across all tab counts and shot settings, and improves

Table 4: Overall closed-world few-shot results after onboarding unseen websites. We report macro ROC-AUC and P@k over *all* monitored websites (base+novel) under different tab counts (3/4/5) and shot budgets (K shots per novel website).

Method (tabs)	5-shot		10-shot		15-shot		20-shot		30-shot	
	AUC	P@k	AUC	P@k	AUC	P@k	AUC	P@k	AUC	P@k
BAPM (3-tab)	0.7416	0.1863	0.7983	0.2712	0.8302	0.3324	0.8490	0.3697	0.8624	0.4007
FMWF (3-tab)	0.8083	0.5509	0.8342	0.5994	0.8472	0.6157	0.8548	0.6294	0.8632	0.6362
TMWF (3-tab)	0.8622	0.5881	0.8918	0.6138	0.8357	0.5989	0.9177	0.6355	0.9336	0.6842
ARES (3-tab)	0.8734	0.6370	0.9109	0.6669	0.9286	0.6932	0.9417	0.7137	0.9629	0.7675
MMF (3-tab)	**0.9453**	**0.6825**	**0.9594**	**0.7302**	**0.9611**	**0.7399**	**0.9667**	**0.7612**	**0.9669**	**0.7824**
BAPM (4-tab)	0.7147	0.2080	0.7697	0.2895	0.7906	0.3353	0.8076	0.3640	0.8218	0.3933
FMWF (4-tab)	0.7746	0.6090	0.8043	0.6524	0.8210	0.6680	0.8272	0.6754	0.8308	0.6774
TMWF (4-tab)	0.8464	0.6773	0.8632	0.6632	0.8628	0.6928	0.8873	0.7094	0.9042	0.6916
ARES (4-tab)	0.8511	0.6884	0.8792	0.6975	0.9018	0.7140	0.9162	0.7293	0.9380	0.7574
MMF (4-tab)	**0.9362**	**0.7189**	**0.9507**	**0.7592**	**0.9572**	**0.7812**	**0.9610**	**0.7923**	**0.9669**	**0.7995**
BAPM (5-tab)	0.6993	0.2460	0.7464	0.3286	0.7645	0.3644	0.7799	0.3939	0.7877	0.4181
FMWF (5-tab)	0.7531	0.6345	0.7947	0.6788	0.8078	0.6790	0.8110	0.6866	0.8125	0.6811
TMWF (5-tab)	0.8393	0.7273	0.8562	0.7410	0.8277	0.7420	0.8847	0.7570	0.9175	0.7821
ARES (5-tab)	0.8352	0.7078	0.8565	0.7125	0.8650	0.7150	0.8811	0.7267	0.9078	0.7409
MMF (5-tab)	**0.9219**	**0.7364**	**0.9384**	**0.7635**	**0.9505**	**0.7869**	**0.9543**	**0.7964**	**0.9584**	**0.8048**

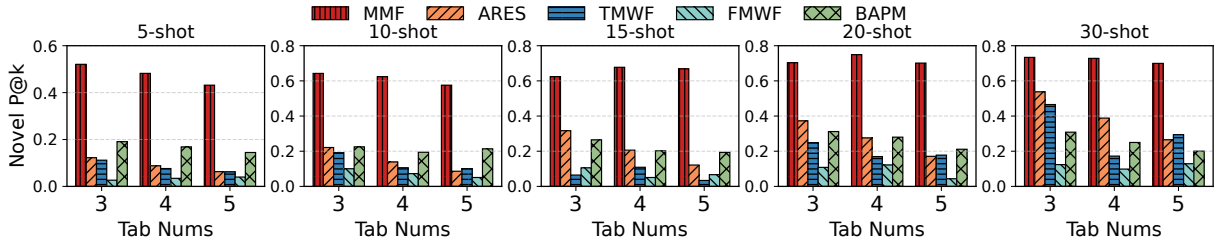


Figure 6: Few-shot onboarding under closed-world multi-tab settings. We report Novel P@k for unseen websites across tab counts (3/4/5 tabs) and shot budgets ($K \in \{5, 10, 15, 20, 30\}$).

steadily with K. With only 20 samples per novel website, MMF achieves a Novel P@k close to 0.8. Moreover, under the most challenging 5-shot/5-tab and 10-shot/5-tab settings, MMF delivers over 300% relative improvement over prior methods. This advantage stems from episodic base training that learns a transferable support-to-reweighting mechanism for extracting class-specific evidence from mixed traces. Moreover, the binary *presence* formulation avoids restructuring a fixed-cardinality classification head when the monitored set expands.

Beyond Novel P@k, Table 4 reports the *overall* AUC and P@k computed over *all* monitored websites (base+novel) after adaptation. Most baselines fail to learn novel classes, yet their overall scores drop less because the metric is dominated by base classes learned during CW training. Even under this metric, MMF achieves the best performance and remains the most stable method across all shot and tab settings.

5.4 Multi-tab Continual Onboarding under Expanding Monitored Set

After validating few-shot adaptation under a fixed novel-set size, we consider a more deployment-realistic setting in which the monitored set *continuously expands* over time. We fix the available samples to 20-shot per unseen website and progressively increase the number of novel classes from 5 to 35. For each setting, we evaluate *novel* website detection quality (Novel P@k), under 3/4/5-tab settings.

As shown in Figure 7, prior baselines degrade steadily as the monitored set grows. Most baselines quickly drop to low Novel P@k and remain there as the monitored set expands. In contrast, MMF remains markedly more scalable and stable across all tab settings. Its Novel P@k consistently stays around 0.75, whereas baselines progressively collapse to 0.1–0.2 under the most challenging 5-tab condition.

Table 5: Few-shot fine-tuning under tab-count mismatch. Models are fine-tuned under a fixed tab-count variant (3/4/5) or mixed-tab variant, and evaluated on the same mixed-tab test set (2–5 tabs). We report Overall ROC-AUC/P@k (base+novel) and Novel P@k.

	3-tab train			4-tab train			5-tab train			mixed-tab train		
	AUC	**P@k**	**Novel P@k**	**AUC**	**P@k**	**Novel P@k**	**AUC**	**P@k**	**Novel P@k**	**AUC**	**P@k**	**Novel P@k**
BAPM	0.8399	0.3599	0.3076	0.8357	0.3251	0.3789	0.8300	0.3333	0.3274	0.8462	0.3821	0.3146
FMWF	0.8303	0.5944	0.1135	0.8187	0.5912	0.1062	0.8157	0.5854	0.0982	0.8468	0.6053	0.1190
TMWF	0.9127	0.6106	0.2394	0.9001	0.6181	0.2663	0.9185	0.6383	0.3176	0.9213	0.6351	0.2481
ARES	0.9311	0.6901	0.3541	0.9271	0.6610	0.3486	0.9135	0.6251	0.2591	0.9388	0.7130	0.3918
MMF	**0.9612**	**0.7761**	**0.7290**	**0.9661**	**0.7892**	**0.8248**	**0.9653**	**0.7871**	**0.8184**	**0.9717**	**0.8111**	**0.7468**

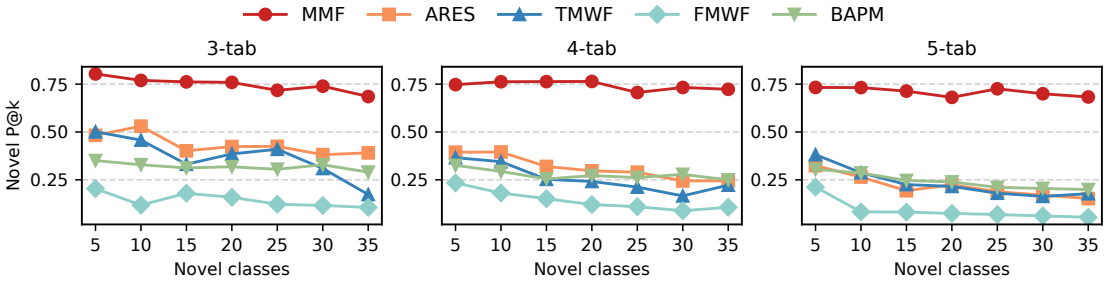


Figure 7: Continual onboarding under an expanding monitored set (20-shot per newly added website). We report Novel P@k as the number of novel classes increases, under 3/4/5-tab mixtures.

5.5 Robustness to Dynamic Tab Settings

As discussed in § 3, the number of concurrently opened tabs is dynamic and typically unknown *a priori*, making it impractical to train and deploy a separate model for each tab count. While some multi-tab attacks attempt to relax this assumption, many prior designs still rely on a fixed tab count or an explicit upper bound to be effective [18, 22, 31]. MMF's decoupled *presence detection* is designed to reduce the reliance on a fixed tab-count configuration. We therefore evaluate MMF under a *dynamic* setting where the test set is a mixed-tab collection consisting of 2/3/4/5-tab traces.

We evaluate few-shot adaptation under tab-count mismatch. At the base-training stage, we train each model under four variants (3-tab, 4-tab, 5-tab, and mixed-tab). At the few-shot stage, we load the corresponding base checkpoints, fine-tune them under the same variant setting, and then evaluate on the mixed-tab test set again. We report Overall AUC/P@k together with Novel P@k in Table 5, and plot novel precision–recall curves in Figure 8 to characterize the precision–recall trade-off induced by the decision threshold.

Table 5 shows that MMF is largely insensitive to tab-count configuration. Even when trained on fixed 3-tab data, MMF achieves an AUC of 0.9612 on the mixed-tab test set, within 0.01 of training directly on mixed-tab data. This observation is consistent with § 5.2, suggesting that MMF's class-conditioned reweighting and presence-detection formulation remains stable under varying tab settings.

Besides, starting from any base checkpoint (3/4/5/mixed), MMF consistently attains the best Overall AUC/P@k and substantially higher Novel P@k than all baselines, indicating that MMF can identify novel classes even when the tab count varies at test time. Figure 8 provides a finer-grained view. Across all four settings, MMF's curves stay closer to the upper-right corner, indicating a better precision–recall trade-off for novel classes. (The curves can appear slightly jagged because they are computed over novel classes only. We do not report PR curves for FMWF due to its low performance in these settings.)

5.6 Robustness to WF Defenses

After evaluating MMF in standard multi-tab and few-shot settings, we further examine its robustness under *website fingerprinting defenses*. Specifically, we adopt datasets generated by two widely-recognized defense mechanisms in the WF community [17, 24], and follow the same few-shot protocol as in § 5.3 (3–5 tabs with 5–30 shots) to assess low-data adaptation under strong obfuscation. We report *Novel P@k* for consistency.

Figure 9 shows that defenses substantially increase the difficulty of few-shot multi-tab WF. All methods suffer pronounced performance drops due to the injected noise and disrupted patterns. Notably, even ARES, which is

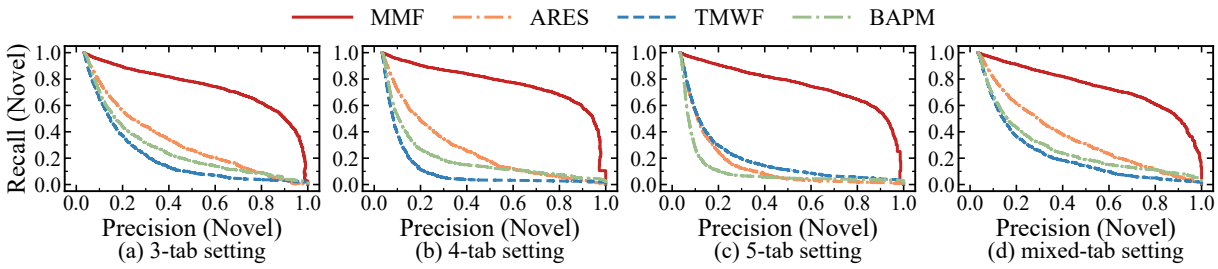


Figure 8: Novel Precision–Recall trade-offs under dynamic tab settings. We plot Novel Precision versus Novel Recall induced by varying threshold, evaluated on the mixed-tab test set (2–5 tabs).

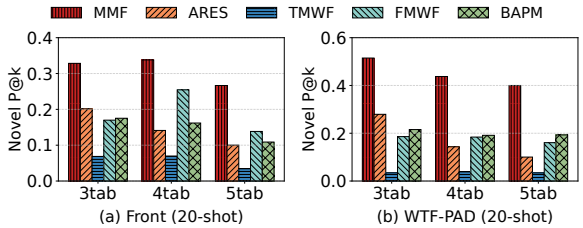


Figure 9: 20-shot performance under defended multi-tab settings. We report Novel P@k across tab counts (3/4/5 tabs).

designed to be robust against WF defenses, fails to remain reliable in this setting, with Novel P@k dropping below 0.25. In contrast, MMF consistently achieves higher and more stable precision under both defenses. This suggests that MMF's decoupled paradigm can relatively better focus on target-related local evidence even when global traffic signatures are heavily distorted. Overall, the results indicate that MMF remains the most robust and practical design among all compared approaches in the low-data and strong-defense settings.

5.7 Robustness to Open-World Scenarios

We evaluate MMF's robustness to background traffic from unmonitored websites and further introduce a distribution shift to emulate deployment dynamics. Concretely, we synthesize open-world traces with $r_{\max} = 0.2$ during base training, but increase it to $r_{\max} = 0.4$ during few-shot fine-tuning, yielding stronger interleaving patterns than those seen during pre-training. As the influence of tab settings has been shown in § 5.3, we only report results in the 4-tab setting here.

As shown in Table 6, MMF remains stable and achieves the best performance under this shift. We further visualize the learned reweighting coefficients for 20 classes in Figure 10. The heatmap shows that different classes activate and suppress markedly different channel combinations. The t-SNE plot shows clear class-wise clustering of the reweighting vectors across multiple inference runs. These observations indicate that MMF learns stable *class-specific* channel patterns rather than being dominated by open-world background noise. During detection, MMF amplifies channels relevant to the target class, whereas spurious patterns introduced by unmonitored traffic lack a corresponding class-conditioned selector and are therefore less likely to produce stable responses.

Table 6: Open-world (OW) robustness under distribution mismatch (4-tab). We report Overall P@k and Novel P@k after few-shot adaptation.

	BAPM	FMWF	TMWF	ARES	MMF
P@k	0.3574	0.6520	0.7063	0.6669	**0.7782**
Novel P@k	0.1775	0.0620	0.1520	0.1343	**0.6236**

5.8 Robustness to Higher Tab Counts and Heavier Interleaving Scenarios

To further explore the influence of tab counts and overlap ratios, we conduct two additional experiments. First, we consider heavier traffic interleaving. As described in § 5.1, our synthesis procedure follows the multi-tab mixing algorithm of TMWF [22], but removes its strict overlap-ratio constraint. The original TMWF mixer computes the overlap window with respect to the shorter of two traces, thereby avoiding cases where a long trace fully subsumes a shorter one. In contrast, our mixer computes the overlap window based on the duration of the previous trace. Thus, when a long trace is followed by a shorter trace, the actual overlap ratio can exceed the configured $r_{\max}$, and the shorter trace can even be entirely contained in the longer one. Figure 11 shows the distribution of actual overlap ratios between adjacent trace pairs under different $r_{\max}$ values. Even when $r_{\max} = 0.4$, 7.01% of adjacent pairs already involve full containment; when $r_{\max} = 0.7$, this fraction increases to 14.80%. We therefore further evaluate the 4-tab setting with $r_{\max} = 0.5$ and

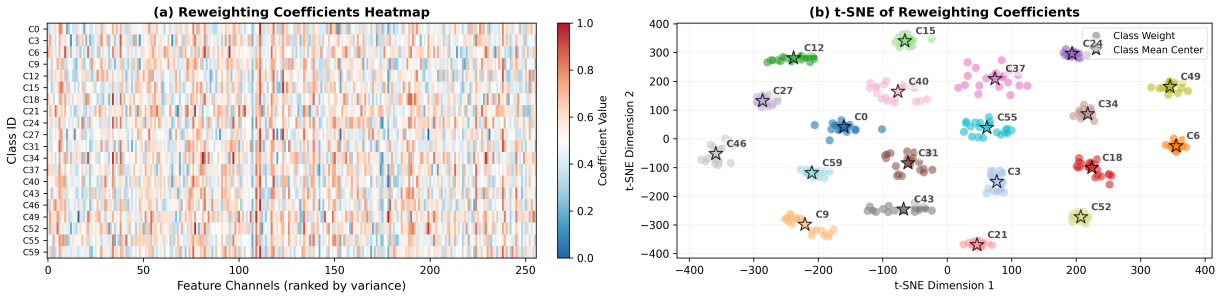


Figure 10: Visualization of class-conditioned reweighting under open-world interference. Left: heatmap of reweighting coefficients over feature channels for 20 classes. Right: t-SNE projection of the corresponding reweighting vectors, showing per-class clustering across multiple inference rounds.

Table 7: Higher tab counts and heavier traffic interleaving robustness. We report Overall ROC-AUC/P@k and Novel P@k after few-shot adaptation (20-shot).

	$r_{\max} = 0.5$,**4-tab**			$r_{\max} = 0.7$,**4-tab**			$r_{\max} = 0.2$,**6-tab**			$r_{\max} = 0.2$,**8-tab**		
	AUC	**P@k**	**Novel P@k**	**AUC**	**P@k**	**Novel P@k**	**AUC**	**P@k**	**Novel P@k**	**AUC**	**P@k**	**Novel P@k**
BAPM	0.8021	0.3593	0.2514	0.7611	0.2943	0.2052	0.7397	0.3829	0.1910	0.6735	0.3364	0.1446
FMWF	0.8228	0.6656	0.1527	0.8123	0.6188	0.1921	0.8023	0.7013	0.0407	0.7503	0.6166	0.0417
TMWF	0.8467	0.6938	0.0988	0.8329	0.6472	0.1160	0.8097	0.8419	0.0550	0.8386	0.8498	0.0479
ARES	0.8292	0.6668	0.0650	0.8105	0.6427	0.0770	0.8467	0.7389	0.0949	0.8026	0.6848	0.0618
MMF	**0.9577**	**0.7818**	**0.7330**	**0.9203**	**0.6958**	**0.6344**	**0.9503**	**0.8734**	**0.7121**	**0.9282**	**0.8587**	**0.6796**

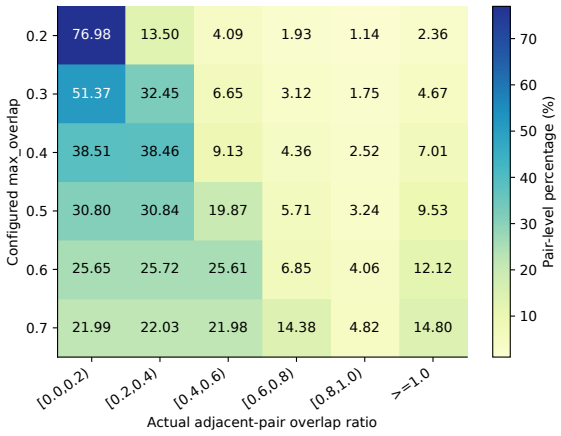


Figure 11: Actual overlap distribution under different interleaving levels in the 4-tab scenario.

$r_{\max} = 0.7$ to test whether MMF remains robust under heavier interleaving.

Second, we evaluate higher tab counts. Prior browser-behavior studies [15, 21, 35] show that the number of simultaneously active tabs follows a long-tailed distribution, and sessions with more than four or five concurrently loading tabs are relatively uncommon. Existing multi-tab WF studies [13, 18, 31] therefore mostly focus on settings with at most five tabs. Nevertheless, MMF performs class-wise presence detection rather than fixed-cardinality combination classification, its prediction process is not inherently tied to a fixed number of concurrent tabs. To examine this property, we additionally construct 6-tab and 8-tab synthetic datasets and evaluate MMF with larger tab counts.

As shown in Table 7, heavier interleaving degrades all methods. This degradation is expected, as denser temporal overlap disrupts more local features and makes target-related patterns harder to separate from mixed traces. Nevertheless, MMF still achieves the best performance. This result suggests that channel reweighting, together with local feature extraction, helps reduce the interference caused by heavily interleaved traffic.

As the number of tabs increases, the fine-tuning traces become increasingly dominated by base classes, because each novel class still appears only in K shots. This makes novel-class adaptation more difficult, especially for strong multi-tab baselines that are primarily optimized to preserve knowledge of base classes. Larger tab counts have a milder impact on MMF than heavier overlap ratios: the Overall P@k decreases by only about 0.015. This indicates that MMF's presence-detection formulation remains relatively stable when more tabs are mixed. Nevertheless, an operational tab-count threshold may still exist for current WF methods, as they commonly rely on fixed-length inputs. Quantitatively identifying this threshold is difficult because the degradation is jointly affected by tab count, trace length and website complexity. We further discuss this limitation in § 6.

Table 8: Multi-tab performance on the real-world collected data. We report Overall P@k and novel P@k in few-shot fine-tuning stage (20shot) under 3/4/5-tab settings.

Method	3-tab		4-tab		5-tab	
	P@k	Novel P@k	P@k	Novel P@k	P@k	Novel P@k
BAPM	0.3067	0.3153	0.2445	0.2488	0.2476	0.1651
FMWF	0.5280	0.3935	0.4270	0.2632	0.4456	0.1252
TMWF	0.5073	0.3992	0.5200	0.3165	0.5396	0.2870
ARES	0.5820	0.4784	0.5360	0.3005	0.5292	0.1421
MMF	**0.6187**	**0.5915**	**0.5960**	**0.5450**	**0.5484**	**0.4702**

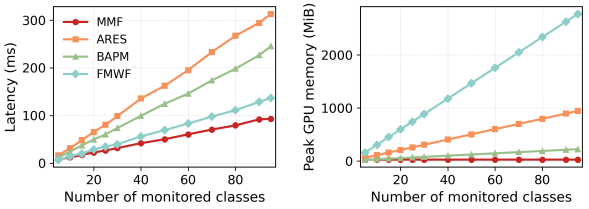


Figure 12: Latency and peak GPU memory as the number of monitored classes increases.

5.9 Real-World Multi-tab Evaluation

To further validate MMF under realistic conditions, we conduct a real-world multi-tab traffic collection study. Considering *concept drift*, we do not attempt to control or eliminate distribution shifts caused by collection time, network conditions, or evolving website content. Instead, we deliberately spread all tab-combination visits across an approximately two-month window, so that training and testing traces inevitably exhibit temporal and environmental mismatch, resembling what an attacker would face when maintaining a deployed model over time.

Table 8 reports the Overall P@k and Novel P@k results for the fine-tuning stage (20-shot). Compared to the controlled setting in § 5.3, all methods suffer a noticeable degradation, with P@k drops exceeding 20%, confirming that concept drift substantially challenges multi-tab WF models, while MMF maintains relatively higher performance. This suggests that existing multi-tab attacks tend to rely on large-scale retraining to recover accuracy under drift, while MMF can remain relatively effective with few-shot adaptation.

5.10 Scalability and Inference Overhead

In practical deployment, the monitored set may grow over time as new websites are added. Therefore, beyond reducing data-collection cost and supporting few-shot onboarding, the inference overhead induced by an expanding monitored set is also an important consideration for practical few-shot multi-tab WF. We conduct a focused scalability analysis by measuring inference latency and peak GPU memory as the number of monitored classes increases.

To the best of our knowledge, ARES [13] is the only existing multi-tab WF framework that explicitly supports website onboarding through a generic one-vs-all architecture. It maintains a separate binary classifier for each monitored class and adds a new classifier whenever a new class is added. Following this deployment setting, we adapt FMWF [31] and BAPM [18] into one-vs-all framework, as they follow the same classification-centric formulation as ARES. We then compare these adapted baselines with MMF.

Figure 12 shows the results. The latency of the one-vs-all framework grows almost linearly with the number of monitored classes, because inference requires forwarding the query through C independent binary classifiers. Their peak GPU memory also increases with the monitored set size, as class-specific models need to be maintained during the inference stage. In contrast, MMF maintains a single shared feature extractor and prediction head. Although MMF applies class-specific reweighting for each monitored class, its overhead is much lower because the query feature map is extracted only once and then reused across class-wise predictions.

MMF's peak GPU memory remains nearly constant as the monitored set grows, since the additional memory required by cached reweighting vectors is much smaller than the cost of maintaining separate binary models. Among the one-vs-all baselines, FMWF shows lower inference latency but higher GPU memory usage than the other adapted baselines. This is likely due to its shorter operator path without attention layers, together with a large flattened fully connected layer. We note that the one-vs-all framework may reduce latency through more aggressive parallelization, at the cost of additional memory and hardware resources. However, such optimization does not remove the underlying scaling issue that the architecture must maintain and execute class-specific classifiers as the monitored set grows.

5.11 Ablation Study

Next, we conduct an ablation study under the 4-tab setting to quantify the contribution of each MMF component. Table 9 shows that replacing any core module substantially degrades performance in the standard multi-tab setting, indicating that these design choices are all important. Specifically, (i) replacing the Trace Encoding module with a simple CNN feature extractor and using an MLP to directly predict the reweighting vector reduces P@k to 0.3134; (ii) replacing the Presence Detection head with a plain MLP head yields an P@k of 0.8423; and (iii) substituting the dynamic reweighting coefficients in Feature Reweighting with a fixed per-class combination

Table 9: Ablation analysis of MMF (4-tab).

Trace Encoding	Feature Reweighting	Presence Detection	P@k
×	✓	✓	0.3134
✓	✓	×	0.8423
✓	×	✓	0.7432
✓	✓	✓	0.9591

representation decreases P@k to 0.7432. These results suggest that MMF's performance arises from the interplay among its components rather than any single module in isolation.

6 Discussion

Research on more practical architectures and benchmarks. MMF advances practical few-shot multi-tab WF by removing the known-tab-count assumption and supporting efficient class updates with limited data. In realistic deployments, the number of concurrent tabs is often unknown, and mismatches between the assumed and actual tab counts are common. Future multi-tab WF studies should therefore evaluate attacks under this setting. However, our work still has two limitations.

First, although MMF achieves lower inference latency and GPU usage than prior onboarding-capable one-vs-all approaches for multi-tab WF [13], the feature reweighting module still introduces additional computation as the monitored set grows. Designing more scalable architectures for large monitored sets remains an important direction for future work.

Second, we do not yet provide a full end-to-end benchmark with human-subject data collection. Such benchmarks remain scarce in the community due to the substantial collection effort involved, as well as the associated ethical and privacy challenges. Building realistic end-to-end benchmarks would be a meaningful step toward more practical WF research.

Difficulty of quantifying a concurrent-tab threshold. The difficulty of identifying a precise tab-count threshold at which multi-tab WF becomes unreliable is closely tied to how current WF pipelines extract features from mixed traces. The feature extractor of current WF methods typically operates on fixed-length inputs, resulting the amount of useful evidence retained depends on the websites' loading behavior and trace length. Our preliminary analysis shows that even with a fixed tab count, performance can vary due to the varying complexity of the mixed websites. Specifically, when using medium-length traces as targets and mixing them with three short or three long traces, the recognition rates are 94.93% and 77.43%, respectively. Therefore, varying only the tab count does not fully isolate the effect of tab concurrency itself. This is a broader limitation of current multi-tab WF evaluation. Future work could address this issue by developing variable-length WF architectures, which may enable a more systematic analysis of how concurrent-tab count affects attack reliability.

Research on concept drift and defenses in multi-tab few-shot settings. Concept drift and defense deployment introduce non-negligible distribution shifts in multi-tab WF, which can materially affect performance. Studying multi-tab WF under these scenarios remains challenging, and, to the best of our knowledge, ARES is one of the few works that explicitly investigates robustness against such factors. This challenge becomes even more pronounced in the few-shot setting. As shown in our evaluation, Front [17] can already substantially reduce the performance of all baseline methods. Meanwhile, recent WF defense research has moved beyond classical lightweight padding schemes, with newer defenses exploring more principled, optimization-driven, and cluster-aware traffic shaping, such as cluster-level traffic anonymization in Palette [38], gradient-guided sequence perturbation in GAPDiS [51], and adaptive per-cluster Tamaraw regularization over preconstructed anonymity sets in Adaptive Tamaraw [26]. Therefore, developing more robust trace representations against WF defenses is a meaningful direction. Moreover, recent robustness studies on traffic classification models, such as Cert-TA [54] and BARS [46], may also provide useful insights for building a systematic framework to understand robustness against different WF defenses in a unified manner.

7 Conclusion

In this paper, we propose MMF, a novel framework for few-shot website fingerprinting in the multi-tab scenario. MMF reformulates WF from classification to class-wise presence detection, which makes episodic learning well-defined under interleaving by pairing each mixed multi-tab query trace with single-tab support traces for monitored websites. MMF encodes each mixed query into a feature map, uses a weight generator to produce class-conditioned reweighting vectors, and applies an attention-guided presence head to detect per-class evidence. This detection-centric formulation avoids requiring tab-count knowledge at inference time and supports continual onboarding of newly added websites from limited newly collected data, without expensive full retraining on large mixed-traffic datasets. Across extensive experiments on public and self-collected datasets, including closed-world and open-world settings, defense-aware evaluation, dynamic tab scenarios, and real-world drift, MMF consistently achieves stronger and more robust performance than prior baselines.

Ethical Considerations

Website fingerprinting (WF) research is inherently dual-use: it can advance the understanding and mitigation of privacy risks in anonymous communication systems, while also potentially enabling misuse for surveillance or de-anonymization.

Stakeholders. Our study may impact: (i) users of anonymous communication systems, whose visited websites could be inferred from encrypted traffic; (ii) the Tor network and its ecosystem, including relay operators, maintainers, and defense developers, who may use our findings to assess practical WF risks and prioritize protocol- or system-level mitigations; (iii) system and defense operators who may use our findings to harden defenses; (iv) website operators/content providers, whose sites may be included in monitored sets in WF evaluations; and (v) the research community, which benefits from more realistic multi-tab benchmarks and reproducible evaluation pipelines.

Positive Impacts. Our primary goal is to *support defense-oriented security research*. We systematically analyze robustness boundaries and help the community evaluate multi-tab WF under more deployment-aligned assumptions.

Rationale for Conducting and Publishing This Work. We conducted this research because the settings in our paper are closer to realistic browsing conditions than the simplified single-tab or data-abundant settings commonly used in prior WF studies. We believe that studying these realistic settings is necessary for improving privacy protections, rather than merely extending attack capability for its own sake. We decided to publish this work because openly characterizing the threat boundary can help Tor developers, anonymity-system researchers, and the broader privacy community identify where existing assumptions may be insufficient and where stronger defenses are needed.

Potential Negative Impacts and Misuse Risks. Despite the above benefits, our technique could be misused to infer real users' browsing activities, which may harm privacy and safety. Moreover, the focus on few-shot adaptation and continual onboarding, while motivated by the practical maintenance challenges of WF, may lower the operational barrier for attackers to keep models up-to-date.

Mitigation Measures (Data Collection & Experimentation). All data used in this work are collected via automated crawling and scripted browsing sessions in a controlled environment. The collection process does not involve real users, user accounts, or any interaction with human subjects; we only record network traces generated by visiting public websites, and our datasets do not include personal identifiers, credentials, or other sensitive user information. We also adopt conservative crawling practices (e.g., bounded concurrency and rate limiting) to avoid imposing abnormal load on websites or anonymity networks, and we do not bypass access controls or collect login-gated content.

Responsible Release and Reproducibility. We follow common practices in the security research community: (1) we position the study as *defense-motivated*, explicitly discussing robustness boundaries and potential countermeasures; and (2) we prioritize *reproducibility* while being cautious about direct weaponization. Consistent with our Open science goal, we provide an anonymized artifact containing the implementation, configurations, and scripts required to reproduce dataset synthesis and experimental results.

Open Science

To support reproducibility and facilitate evaluation of our core contributions, we provide public research artifacts that contain the source code, configuration files, and scripts required to reproduce our dataset synthesis and experimental results. The artifacts are available on Zenodo[1] and GitHub[2]. The repositories include instructions for environment setup, running base training and few-shot fine-tuning, and reproducing the reported metrics and figures.

Acknowledgement

We thank our shepherd and the anonymous reviewers for their insightful feedback. This work was supported in part by the National Key R&D Program of China under Grant 2024YFB2906803, the National Natural Science Foundation of China (NSFC) under Grant 62472247, the Key Research and Development Project of Jiangsu Province under Grant BE2023004-4, and the Science and Technology Innovation Program of Hunan Province under Grant 2024RC3139.

References

[1] Kota Abe and Shigeki Goto. Fingerprinting attack on tor anonymity using deep learning. *Proceedings of the Asia-Pacific Advanced Network*, 42(0):15–20, 2016.

[2] Amit Alfassy, Leonid Karlinsky, Amit Aides, Joseph Shtok, Sivan Harary, Rogerio Feris, Raja Giryes, and Alex M Bronstein. Laso: Label-set

[1] Available at https://zenodo.org/records/20416451.
[2] Available at https://github.com/nl77-seraph/MMF.

operations networks for multi-label few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6548–6557, 2019.

[3] Alireza Bahramali, Ardavan Bozorgi, and Amir Houmansadr. Realistic website fingerprinting by augmenting network traces. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1035–1049, 2023.

[4] Sanjit Bhat, David Lu, Albert Kwon, and Srinivas Devadas. Var-cnn: A data-efficient website fingerprinting attack based on deep learning. *Proceedings on Privacy Enhancing Technologies*, 2019:292 – 310, 2018.

[5] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. Touching from a distance: website fingerprinting attacks and defenses. In *Computer and Communications Security*, 2012.

[6] Hao Chen, Yali Wang, Guoyou Wang, and Yu Qiao. Lstd: A low-shot transfer detector for object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[7] Mantun Chen, Yongjun Wang, and Xiatian Zhu. Few-shot website fingerprinting attack with meta-bias learning. *Pattern Recognition*, 130:108739, 2022.

[8] Heyning Cheng and Ron Avnur. Traffic analysis of ssl encrypted web browsing. *Project paper, University of Berkeley*, 1998.

[9] Giovanni Cherubin, Rob Jansen, and Carmela Troncoso. Online website fingerprinting: Evaluating website fingerprinting attacks on tor in the real world. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 753–770, 2022.

[10] Weiqi Cui, Tao Chen, Christian Fields, Julianna Chen, Anthony Sierra, and Eric Chan-Tin. Revisiting assumptions for website fingerprinting attacks. In *Proceedings of the 2019 ACM asia conference on computer and communications security*, pages 328–339, 2019.

[11] Xianwen Deng, Ruijie Zhao, Yanhao Wang, Mingwei Zhan, Zhi Xue, and Yijun Wang. Countmamba: A generalized website fingerprinting attack via coarse-grained representation and fine-grained prediction. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 1419–1437. IEEE, 2025.

[12] Xinhao Deng, Qi Li, and Ke Xu. Robust and reliable early-stage website fingerprinting attacks via spatial-temporal distribution analysis. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 1997–2011, 2024.

[13] Xinhao Deng, Xiyuan Zhao, Qilei Yin, Zhuotao Liu, Qi Li, Mingwei Xu, Ke Xu, and Jianping Wu. Towards robust multi-tab website fingerprinting. *arXiv preprint arXiv:2501.12622*, 2025.

[14] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, 2004.

[15] Patrick Dubroy and Ravin Balakrishnan. A study of tabbed browsing among mozilla firefox users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 673–682, 2010.

[16] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[17] Jiajun Gong and Tao Wang. Zero-delay lightweight defenses against website fingerprinting. In *29th USENIX security symposium (USENIX security 20)*, pages 717–734, 2020.

[18] Zhong Guan, Gang Xiong, Gaopeng Gou, Zhen Li, Mingxin Cui, and Chang Liu. Bapm: block attention profiling model for multi-tab website fingerprinting attacks on tor. In *Proceedings of the 37th Annual Computer Security Applications Conference*, pages 248–259, 2021.

[19] Jamie Hayes and George Danezis. k-fingerprinting: A robust scalable website fingerprinting technique. In *USENIX Security Symposium*, 2015.

[20] Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath. Website fingerprinting: Attacking popular privacy enhancing technologies with the multinomial nave-bayes classifier. *ACM*, 2009.

[21] Jeff Huang and Ryen W White. Parallel browsing behavior on the web. In *Proceedings of the 21st ACM conference on Hypertext and Hypermedia*, pages 13–18, 2010.

[22] Z. Jin, T. Lu, S. Luo, and J. Shang. Transformer-based model for multi-tab website fingerprinting attack. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2023.

[23] Marc Juárez, Sadia Afroz, Gunes Acar, Claudia Díaz, and Rachel Greenstadt. A critical evaluation of website fingerprinting attacks. *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014. URL: https://api.semanticscholar.org/CorpusID:5762455.

[24] Marc Juárez, Mohsen Imani, Mike Perry, Claudia Dıaz, and Matthew Wright. Wtf-pad: toward an efficient website fingerprinting defense for tor. *CoRR, abs/1512.00524*, 2015.

[25] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[26] Khashayar Khajavi and Tao Wang. Lightening the load: A cluster-based framework for a lower-overhead, provable website fingerprinting defense. *arXiv preprint arXiv:2509.01046*, 2025.

[27] Lin Liu, Ziling Wei, Shuhui Chen, Zixuan Dong, and Jinshu Su. Fingermamba: Mamba-based efficient multi-tab website fingerprinting. In *2024 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pages 1–9. IEEE, 2024.

[28] Lin Liu, Ziling Wei, Shuhui Chen, Jinshu Su, and Yong Yuan. Kp-wf: Cross-domain few-shot website fingerprinting. In *International Conference on Digital Forensics and Cyber Crime*, pages 227–245. Springer, 2024.

[29] Chenxiang Luo, Wenyi Tang, Qixu Wang, and Danyang Zheng. Few-shot website fingerprinting with distribution calibration. *IEEE Transactions on Dependable and Secure Computing*, 22(1):632–648, 2024.

[30] Akshaya Mani, T Wilson-Brown, Rob Jansen, Aaron Johnson, and Micah Sherr. Understanding tor usage with privacy-preserving measurement. In *Proceedings of the Internet Measurement Conference 2018*, pages 175–187, 2018.

[31] Wenwen Meng, Chuan Ma, Ming Ding, Chunpeng Ge, Yuwen Qian, and Tao Xiang. Beyond single tabs: A transformative few-shot approach to multi-tab website fingerprinting attacks. In *Proceedings of the ACM on Web Conference 2025*, pages 1068–1077, 2025.

[32] Andriy Panchenko, Fabian Lanze, Jan Pennekamp, Thomas Engel, Andreas Zinnen, Martin Henze, and Klaus Wehrle. Website fingerprinting at internet scale. In *Network and Distributed System Security Symposium*, 2016. URL: https://api.semanticscholar.org/CorpusID:15302617.

[33] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. Website fingerprinting in onion routing based anonymization networks. In *Proceedings of the 10th ACM Workshop on Privacy in the Electronic Society (WPES)*, 2011.

[34] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International conference on learning representations*, 2017.

[35] Daisy Reyes, Eno Dynowski, Taryn Chovan, John Mikos, Eric Chan-Tin, Mohammed Abuhamad, and Shelia Kennison. Webtracker: Real webbrowsing behaviors. In *2023 Silicon Valley Cybersecurity Conference (SVCC)*, 2023.

[36] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen. Automated website fingerprinting through deep learning. In *Network and Distributed System Security Symposium*, 2017.

[37] Vera Rimmer, Theodor Schnitzler, Tom Van Goethem, Abel Rodríguez Romero, Wouter Joosen, and Katharina Kohls. Trace oddity: Methodologies for data-driven traffic analysis on tor. *Proceedings on Privacy Enhancing Technologies*, 2022(3):314–335, 2022.

[38] Meng Shen, Kexin Ji, Jinhe Wu, Qi Li, Xiangdong Kong, Ke Xu, and Liehuang Zhu. Real-time website fingerprinting defense via traffic cluster anonymization. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 3238–3256. IEEE, 2024.

[39] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. Deep fingerprinting: Undermining website fingerprinting defenses with deep learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1928–1943, 2018.

[40] Payap Sirinam, Nate Mathews, Mohammad Saidur Rahman, and Matthew K. Wright. Triplet fingerprinting: More practical and portable website fingerprinting with n-shot learning. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019. URL: https://api.semanticscholar.org/CorpusID:207960329.

[41] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[42] Jingwen Tan, Huanran Wang, Shuai Han, Dapeng Man, and Wu Yang. An adaptability-enhanced few-shot website fingerprinting attack based on collusion. *IEEE Transactions on Information Forensics and Security*, 2024.

[43] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Random k-labelsets for multilabel classification. *IEEE transactions on knowledge and data engineering*, 23(7):1079–1089, 2010.

[44] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

[45] David Wagner, Bruce Schneier, et al. Analysis of the ssl 3.0 protocol. In *The Second USENIX Workshop on Electronic Commerce Proceedings*, volume 1, pages 29–40, 1996.

[46] Kai Wang, Zhiliang Wang, Dongqi Han, Wenqi Chen, Jiahai Yang, Xingang Shi, and Xia Yin. Bars: Local robustness certification for deep learning based traffic analysis systems. In *NDSS*, 2023.

[47] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson, and Ian Goldberg. Effective attacks and provable defenses for website fingerprinting. *USENIX Security Symposium*, 2014.

[48] Tao Wang and Ian Goldberg. Improved website fingerprinting on tor. In *Workshop on Privacy in the Electronic Society*, 2013.

[49] Tao Wang and Ian Goldberg. On realistically attacking tor with website fingerprinting. *Proceedings on Privacy Enhancing Technologies*, 2016.

[50] Zhiquan Wang, Tengyao Li, Meijuan Yin, Xiaoyun Yuan, Xiangyang Luo, and Lingling Li. Wf3a: A n-shot website fingerprinting with effective fusion feature attention. *Computers & Security*, 140:103796, 2024.

[51] Ruotian Xie, Kun Xie, Pengcheng Zhao, Jiajun He, Xin Zeng, Jigang Wen, Yong Xie, Wei Liang, and Gaogang Xie. Gapdis: Gradient-assisted perturbation design via sequence editing for website fingerprinting defense. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pages 3012–3026, 2025.

[52] Yi Xie, Jiahao Feng, Wenju Huang, Yixi Zhang, Xueliang Sun, Xiaochou Chen, and Xiapu Luo. Contrastive fingerprinting: A novel website fingerprinting attack over few-shot traces. In *Proceedings of the ACM Web Conference 2024*, pages 1203–1214, 2024.

[53] Yixiao Xu, Tao Wang, Qi Li, Qingyuan Gong, Yang Chen, and Yong Jiang. A multi-tab website fingerprinting attack. In *Proceedings of the 34th Annual Computer Security Applications Conference*, pages 327–341, 2018.

[54] Jinzhu Yan, Zhuotao Liu, Yuyang Xie, Shiyu Liang, Lin Liu, and Ke Xu. {CertTA}: Certified robustness made practical for {Learning-Based} traffic analysis. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 7349–7368, 2025.

[55] Qilei Yin, Zhuotao Liu, Qi Li, Tao Wang, Qian Wang, Chao Shen, and Yixiao Xu. An automated multi-tab website fingerprinting attack. *IEEE Transactions on Dependable and Secure Computing*, 19(6):3656–3670, 2021.

[56] Yali Yuan, Weiyi Zou, and Guang Cheng. Mw3f: Improved multi-tab website fingerprinting attacks with transformer-based feature fusion. *Journal of Network and Computer Applications*, 236:104125, 2025.

[57] Gongjie Zhang, Zhipeng Luo, Kaiwen Cui, Shijian Lu, and Eric P Xing. Meta-detr: Image-level few-shot detection with inter-class correlation exploitation. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):12832–12843, 2022.

[58] Xiyuan Zhao, Xinhao Deng, Qi Li, Yunpeng Liu, Zhuotao Liu, Kun Sun, and Ke Xu. Towards fine-grained webpage fingerprinting at scale. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 423–436, 2024.

[59] Qiang Zhou, Liangmin Wang, Huijuan Zhu, and Tong Lu. Few-shot website fingerprinting attack with cluster adaptation. *Computer Networks*, 229:109780, 2023.

[60] Hongcheng Zou, Jinshu Su, Ziling Wei, Shuhui Chen, Chunfang Yang, and Mantun Chen. Toward an effective few-shot website fingerprinting attack with quadruplet networks and deep local fingerprinting features. *IEEE Transactions on Dependable and Secure Computing*, 2025.

Algorithm 1: Time-consistent mixing for synthesizing an n-tab query trace.

Require: n single-tab traces $\{T_j\}_{j=1}^{n}$, overlap ratio range $[r_{\min}, r_{\max}]$
Ensure: Mixed query direction sequence x
1: Each trace $T_j = \{(t_{j,m}, d_{j,m})\}_{m=1}^{n_j}$, where $d_{j,m} \in \{+1, -1\}$.
2: Normalize timestamps in each trace so that $t_{j,1} = 0$.
3: $\text{start}_1 \leftarrow 0$.
4: **for** $j = 2$ to n **do**
5: Sample $r_j \sim \mathcal{U}(r_{\min}, r_{\max})$.
6: $D_{j-1} \leftarrow$ duration of T_{j-1}.
7: $\Delta \leftarrow r_j \cdot D_{j-1}$ {overlap measured on the previous tab}
8: $\text{start}_j \leftarrow \text{start}_{j-1} + D_{j-1} - \Delta$.
9: Shift timestamps in T_j: $t_{j,m} \leftarrow t_{j,m} + \text{start}_j \quad \forall m$.
10: **end for**
11: $E \leftarrow$ merge all events from $\{T_j\}$ and sort by timestamp.
12: $x \leftarrow$ the direction sequence $\{d\}$ extracted from E in temporal order.
13: **return** x

Table 10: Data Collection Environment Specifications

Component	Specification
Operating System	Ubuntu 22.04 LTS (Dockerized)
Browser	Tor Browser 13.5 (Standard Security Level)
Automation Framework	Python Selenium with Geckodriver
Traffic Capture	tcpdump
Display Environment	Xvfb (Headless X Server)
Network Strategy	Default Tor circuit construction (Random Guard/Exit)

A Detailed Settings For Experiment

A.1 Multi-tab synthesis procedure.

To emulate realistic multi-tab browsing, we synthesize an n-tab query trace by *time-consistent mixing* of multiple labeled single-tab traces (Algorithm 1). We use time-based composition (event-level merging) instead of sequential concatenation because it preserves bursty micro-structures that are important for WF. Each single-tab trace is represented as timestamped packet directions, where an outgoing packet is encoded as $+1$ and an incoming packet as -1. Given n traces, we normalize their timestamps to start from zero and assign a start time to each trace so that tab openings partially overlap. For $j \geq 2$, we sample an overlap ratio $r_j \sim \mathcal{U}(r_{\min}, r_{\max})$ and set

$$\text{start}_j = \text{start}_{j-1} + (1 - r_j) D_{j-1},$$

where D_{j-1} is the duration of the previous trace. Equivalently, the intended overlap window between T_{j-1} and T_j is $\Delta_j = r_j \cdot D_{j-1}$, i.e., T_j begins Δ_j time units before T_{j-1} ends. This construction naturally allows *containment* when a previous tab is much longer than a newly opened tab: if $D_j \leq r_j D_{j-1}$, then T_j can be fully contained within T_{j-1}'s time span. In extreme cases, a long-lived tab may contain multiple subsequently opened short tabs, matching real sessions. After shifting timestamps, we merge all packets and extract the mixed direction sequence in temporal order.

During base training, we split single-tab data into training and test sets with an 8:2 ratio and then the synthesis algorithm is applied to ensure strictly separation. For the few-shot stage, to mitigate evaluation bias under limited onboarding samples, we use a more stringent split where the test set is five times larger than the training set, improving result stability.

In the closed-world (CW) setting, we mix only monitored websites and use a larger overlap range (e.g., $r_{\max} = 0.4$) to induce stronger interleaving. In the open-world (OW) setting, we additionally insert an unmonitored trace at a random position in the tab combination.

A.2 Data Collection Infrastructure and Protocol

We improved the data collection framework based on the method provided by ARES [13]. The collection pipeline is designed to simulate a realistic passive adversary while maintaining strict experimental controls.

Infrastructure Configuration. We utilized Docker containers to isolate browsing instances, allowing for parallel data collection without cross-talk. Each container was allocated independent system resources to prevent CPU throttling artifacts in the traffic traces. The key specifications of our collection environment are detailed in Table 10.

Collection Protocol. Our collection script orchestrates the browsing sessions following a strict protocol to ensure data quality:

- **Session Isolation:** For every collected trace (instance), a fresh Tor Browser profile is created. This ensures that no cache, cookies, or local storage artifacts persist between visits, guaranteeing statistical independence of the samples.
- **Traffic Capture:** We capture traffic at the container's network interface using `tcpdump`. To isolate the browsing traffic from background noise, we filter strictly for TCP packets.
- **Verification:** To verify successful page loads, we capture a screenshot of the virtual framebuffer (via Xvfb). Traces where the screenshot indicates a loading error (e.g., Tor connection timeout) are discarded.

We will publicly release all code and configuration related to data collection to support reproducibility and future research.