Reproducing Deep Fingerprinting: A Replication Study of Deep Learning-Based Website Fingerprinting Attacks

Devin Fung
dfung4@stanford.edu
Stanford University

Miro Ohannes Swisher
mswisher@stanford.edu
Stanford University

Yousef Belal Helal
yousefh@stanford.edu
Stanford University

Deepashree Kedia
deepkd@stanford.edu
Stanford University

Abstract

Website fingerprinting attacks infer a user's browsing activity from patterns in encrypted network traffic, even over anonymity networks like Tor. We replicate the Deep Fingerprinting (DF) attack of Sirinam et al. (2018), which uses a 1D convolutional neural network trained on raw packet direction sequences, and extend its evaluation to two contemporary low-latency defenses (RegulaTor and BRO), alongside the original paper's defenses. Our replication achieves 96.1% closed-world accuracy on undefended traffic. Against modern defenses, accuracy varies widely: RegulaTor and BRO reduce accuracy to 81.8% and 74.1% respectively, while high-overhead defenses BuFLO and Tamaraw reduce it to 31.7% and 26.1%. Open-world results follow a similar pattern. We also independently collect 91,955 Tor traffic traces in 2026 and discover a critical representation pitfall: capturing TCP packets instead of Tor cells fundamentally changes the data distribution. After correcting this, our self-collected data achieves only 48.0% accuracy, and cross-dataset experiments show ∼1% accuracy in both directions, demonstrating that DF learns distribution-specific patterns rather than universal website fingerprints.

1 Introduction

Tor [2] is the most widely deployed anonymity network, used by millions of people to avoid surveillance and censorship. While Tor encrypts traffic and routes it through multiple relays, a local passive adversary at the user's entry guard can still observe packet timing, size, and direction. From these observable features alone, an attacker can identify which website a user is visiting, a *website fingerprinting* (WF) attack.

Early WF attacks relied on hand-crafted features fed to classical classifiers [4]. Sirinam et al.'s 2018 Deep Fingerprinting paper [8] changed the picture: a 1D CNN trained directly on packet direction sequences achieved over 98% closed-world accuracy with no feature engineering, and remained effective against WTF-PAD, a defense that had previously been considered strong.

That result is now several years old. Newer defenses, RegulaTor [5] and BRO [7], were designed with low latency as a hard constraint and with awareness of deep learning–based attacks. Whether DF still poses a credible threat against these defenses is an open question. We make three key contributions:

1. We replicate the original DF setup and confirm reproducibility, recovering 96.1% closed-world accuracy on undefended traffic.
2. We evaluate DF against seven defenses in total: the original paper's WTF-PAD and Walkie-Talkie, plus RegulaTor, BRO, BuFLO [3], and Tamaraw [1]. We report both closed-world accuracy and open-world precision/recall for all.
3. We independently collect 91,955 Tor traffic traces in 2026 and test whether DF's learned fingerprints generalize across time and collection environment. They do not: a model trained on the original 2018 data achieves only ∼1% accuracy on our traces, indistinguishable from random chance.

2 Background and Related Work

2.1 Website Fingerprinting

In a WF attack, the adversary is a local passive eavesdropper, an ISP or on-path observer, who can see packet sizes, timing, and direction but not payload content. The closed-world setting restricts classification to a fixed set of monitored sites; the open-world setting adds a large pool of unmonitored sites, requiring the attack to also reject traffic that does not match any monitored class.

2.2 Deep Fingerprinting

The DF model [8] takes raw packet direction sequences as input (each packet encoded +1 for outgoing, −1 for incoming, zero-padded to 5,000) and passes them through a four-block 1D CNN. The first block uses ELU activations to preserve the sign of incoming packets; later blocks use ReLU. A two-layer fully connected classifier

follows the convolutional backbone. Sirinam et al. reported 98.3% closed-world accuracy on 95 monitored sites with no defense.

2.3 Traffic Shaping Defenses

Traffic-shaping defenses attempt to reduce fingerprintability by modifying observable flow characteristics, such as packet timing and transmission rate. Of the defenses we evaluate, RegulaTor and BRO were introduced after the original paper's release. We evaluate the following defenses:

WTF-PAD [6] inserts dummy packets adaptively to disrupt timing patterns, with low bandwidth overhead.

WalkieTalkie [9] reshapes traffic into half-duplex constant-rate bursts. Evaluations typically report top-2 accuracy because the number of bursts is still a distinguishing feature.

BuFLO [3] sends at a constant rate in both directions for a fixed duration, providing strong obfuscation at significant overhead.

Tamaraw [1] extends BuFLO with separate send rates per direction, offering stronger privacy guarantees at similar cost.

RegulaTor [5] regulates per-direction send rates toward a target, prioritizing low latency while still substantially reshaping the traffic pattern.

BRO [7] is a recent low-overhead defense designed with explicit awareness of deep learning–based attackers.

3 Approach and Implementation

3.1 Architecture

Figures 1 and 2 show the experimental pipelines for the public and self-collected datasets respectively.

We implement the DF architecture exactly as described in Sirinam et al., with no modifications. The model takes a 5,000-element direction sequence and outputs a softmax over monitored classes.

The backbone has four convolutional blocks. Block 1 uses 32 filters with ELU activations; Blocks 2–4 use 64, 128, and 256 filters with ReLU. Within each block, two convolutions (kernel size 8) are applied, each followed by batch normalization and activation, then a max-pool layer (size 8, stride 4) and dropout (rate 0.1). The classifier consists of two dense layers of width 512, each regularized with batch normalization and dropout (rates 0.7 and 0.5), followed by the final softmax. All weights use Glorot uniform initialization with a fixed seed.

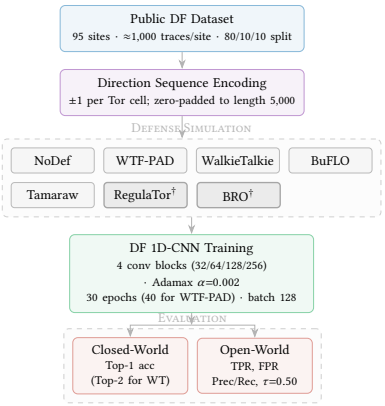

Figure 1. Public DF pipeline on 5,000-length ±1 direction sequences under seven defenses. RegulaTor and BRO (†) are post-2018 low-latency defenses; WalkieTalkie open-world is unavailable.

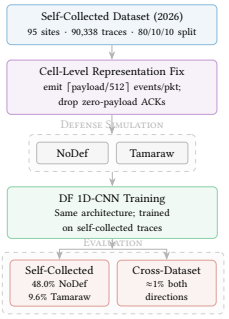

Figure 2. Self-collected pipeline (March 2026): fixing TCP-packet vs. Tor-cell counting raises accuracy from 37% to 48%. Only NoDef and Tamaraw are evaluated, and cross-dataset transfer is ≈1% in both directions.

3.2 Training Procedure

We use the Adamax optimizer ($\alpha = 0.002$, $\beta_1 = 0.9$, $\beta_2 = 0.999$) with categorical cross-entropy loss and a batch size of 128. Training runs for 30 epochs for most conditions and 40 epochs for WTF-PAD, matching the original paper. All experiments use a fixed random seed.

3.3 Data Representation

Each trace is a sequence of packet directions: +1 (client to guard) or −1 (guard to client), truncated or zero-padded to 5,000 elements. We use direction only without size, due to the original papers findings that adding packet lengths didn't noticeably improve accuracy [8]. The fixed length of 5,000 is chosen to cover the vast majority of traces without excessive zero-padding. Sirinam et al. report that only 8,121 of 95,000 closed-world traces exceeded 5,000 cells and required truncation, and

the remainder were shorter and were right-padded with zeros.

3.4 Closed-World Evaluation

In the closed-world setting, we assume that every trace in the test set belongs to one of the monitored sites. The classifier's task is therefore a standard multi-class classification problem: given an input trace, assign it to exactly one of the 95 monitored classes.

We evaluate closed-world performance using top-1 accuracy: the fraction of test traces for which the model's prediction matches the true label. For WalkieTalkie, we additionally report top-2 accuracy, because the defense's collision mechanism molds pairs of sites to look identical. Top-2 captures whether the model correctly narrowed the answer to the right pair even if it picked the wrong member.

3.5 Open-World Evaluation

The closed-world setting assumes the user can only visit one of the monitored sites. The vast majority of websites on the interne, however, are not in the attacker's monitored set. The open-world setting captures this reality by making it so that the classifier must decide not only which monitored site a trace belongs to, but also whether the trace belongs to any monitored site at all.

We apply the closed-world model to a mixed test set containing both monitored and unmonitored traces. Following the original papers guidance, we train the classifier on both monitored and unmonitored traces (all belonging to a single unmonitored class), which teaches the model what unmonitored traces look like.

The model then outputs a softmax probability distribution over the monitored classes. We then classify a trace as monitored if the maximum softmax probability over monitored classes exceeds the threshold τ (which measures how "confident" the model is about a positive identification); otherwise it is rejected as unmonitored. We report TPR, FPR, precision, and recall at $\tau = 0.50$.

4 Experimental Setup and Dataset

4.1 Dataset

Baseline. Our primary replication uses the publicly available dataset from the original DF paper: 95 monitored sites with approximately 1,000 traces each, plus a large unmonitored set, split 80/10/10 for train/validation/test.

Defended conditions. WTF-PAD, WalkieTalkie, BuFLO, RegulaTor, BRO, and Tamaraw were applied as simulated defenses to the same original-paper traces, preserving comparability with the NoDef baseline. we were not able to capture WalkieTalkie defense data for our open-world evaluation due to time constraints.

Self-collected dataset. We also collected our own Tor traffic for the same Alexa Top-100 sites in March 2026. Our setup uses headless Firefox driven by Selenium, routed through Tor's SOCKS proxy, with tcpdump recording all traffic to and from the entry guard. We ran this on two machines: a local workstation (WSL2, 15 parallel Tor workers) and a Google Cloud VM (e2-standard-16, 30 workers), collecting 91,955 pcap files in total. We ended up with 90,338 usable traces across 95 classes (5 sites were dropped for insufficient data), split 80/10/10. Notably, our traces are much shorter than the benchmark's: a median of 1,135 non-zero cells versus 4,022, likely reflecting how much website loading behavior has changed between 2016–2018 and 2026.

4.2 Representation Bug: Packets vs. Cells

Our initial experiments on self-collected data produced only 37% accuracy. Investigation revealed a representation mismatch: the DF paper uses Tor *cell-level* sequences (512-byte cells), but our `tcpdump` pipeline counted every TCP packet, including zero-payload ACKs, as a direction event, inflating the outgoing ratio from the expected ∼15% to ∼45%. This bug affected all three independent collection efforts in our group. Our fix skips zero-payload packets and emits $\lceil \texttt{payload}/512 \rceil$ direction events per data-carrying packet, improving accuracy to 48.0%.

5 Evaluation

5.1 Closed-World Accuracy

Table 1 shows top-1 accuracy on the held-out test split for each defense condition, with the model being trained and tested on the original paper's data. For WalkieTalkie we additionally reports top-2 accuracy, due to the nature of its defense mechanism. Figure 3 also demonstrates the accuracy of each defense as the model was trained.

Our NoDef accuracy of 96.1% is close to, but slightly below, the 98.3% reported by Sirinam et al.

5.2 Open-World Results

Table 2 reports open-world performance at $\tau = 0.50$, while Figure 3 shows representative training dynamics across defenses. WalkieTalkie open-world results are not available.

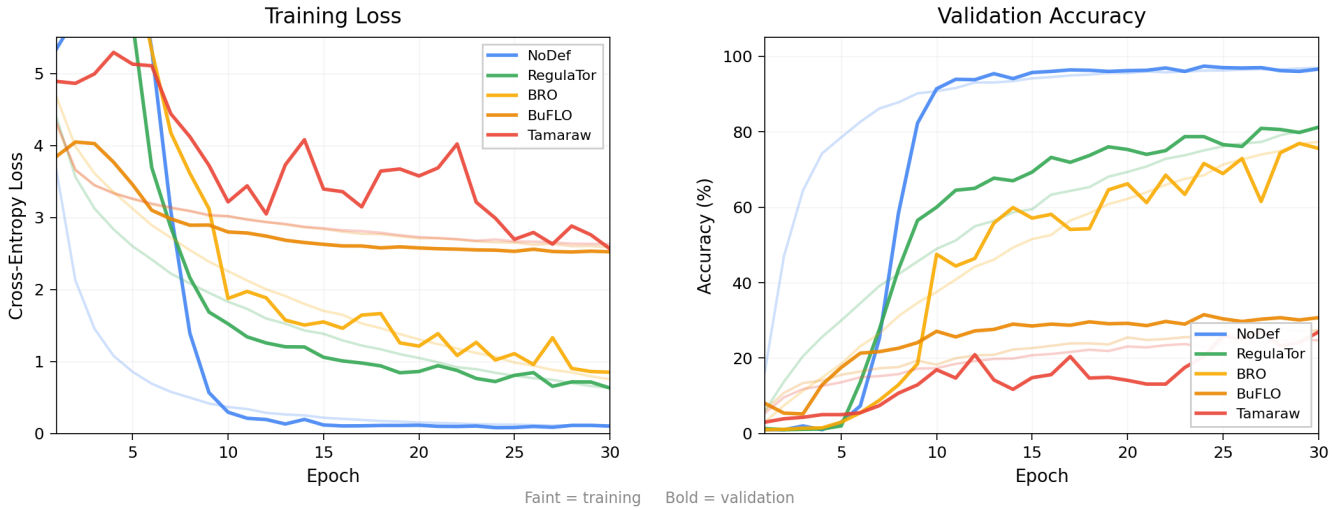

Figure 3. Training and validation behavior across epochs for selected defense conditions. Convergence is stable for NoDef and RegulaTor, while stronger defenses exhibit lower validation accuracy throughout training.

Table 1. Closed-world classification accuracy.

Defense	Top-1 Accuracy (Top-2)
NoDef	0.961
RegulaTor	0.818
WTF-PAD	0.895
BRO	0.741
WalkieTalkie	0.461 (top-2: 0.737)
BuFLO	0.317
Tamaraw	0.261

Table 2. Open-world performance at $\tau = 0.50$.

Defense	TPR	FPR	Precision	Recall
NoDef	0.9895	0.0010	0.9989	0.9895
RegulaTor	0.8189	0.0080	0.9898	0.8189
BRO	0.7284	0.0680	0.9105	0.7284
BuFLO	0.1432	0.0460	0.7473	0.1432
Tamaraw	0.0747	0.0250	0.7396	0.0747
WalkieTalkie	[results not available]			

5.3 Self-Collected Results

Training and testing the model on our cell-level self-collected data, DF achieves 48.0% closed-world accuracy on 95 classes, well above random (1.05%), but far below the benchmark's 96.1%. The remaining gap reflects shorter traces, temporal distribution shift (2026 vs. 2016–2018 websites with different loading patterns), and our approximate cell counting (28.8% vs. 15.5% outgoing ratio). Applying the Tamaraw defense simulator reduced accuracy to 9.6%, nearly random, confirming that constant-rate padding is effective even on independently collected, noisier traffic.

5.4 Cross-Dataset Generalization

We trained DF on each dataset and evaluated on the other (Table 3).

Table 3. Cross-dataset accuracy (cell-level).

Direction	In-Dist.	Cross
Benchmark → Ours	97.9%	1.24%
Ours → Benchmark	47.4%	1.37%

Cross-dataset accuracy is ~1% in both directions, random chance, even after the cell-level fix. This demonstrates that DF learns *distribution-specific* patterns (guard timing, network TCP behavior, etc.) rather than universal website fingerprints. An attacker thus cannot train DF from one environment and expect it to generalize without retraining, substantially limiting the real-world threat.

6 Discussion

Our results replicate the core finding of Sirinam et al.: a 1D CNN trained on raw packet direction sequences is

a strong attacker against Tor traffic. The 96.1% closed-world accuracy we recover on undefended traffic confirms that the DF model is reproducible and that the original claims hold.

The more interesting signal is in the defense results. RegulaTor and BRO were designed with low latency as a hard constraint and, in BRO's case, with explicit knowledge that deep-learning classifiers are the adversary. Yet DF still achieves 81.8% and 74.1% accuracy against them. This suggests that rate-shaping defenses that preserve the general shape of the traffic pattern leave enough residual signal for a CNN to exploit. Padding-heavy defenses tell a different story: BuFLO and Tamaraw push accuracy below 32%, consistent with the intuition that replacing the true traffic pattern with a near-constant bitrate effectively destroys the features the model relies on.

The open-world results reinforce this picture. Precision remains high even for weak defenses, above 0.99 for NoDef and RegulaTor, meaning the model rarely raises false alarms on unmonitored traffic when it does fire. TPR degrades much faster than precision as defenses strengthen, which is the expected behavior: the model becomes conservative rather than noisy.

There is reason to believe the limitation for low-latency constrained defenses such as BRO and RegulaTor is at least partly fundamental. Low-overhead defenses, by definition, cannot pad or delay traffic enough to fully obscure the underlying pattern. RegulaTor reshapes send rates toward a target but still transmits the same total volume of data in roughly the same order. BRO adds obfuscation but keeps overhead low enough for practical deployment. A CNN with sufficient depth can learn to "see through" moderate perturbations, and an adaptive defense tuned to target DF's convolutional layers could potentially achieve much stronger protection at the same overhead budget.

Our self-collected data paints a more nuanced picture of DF's practical threat. When we train DF from scratch on our own 2026 traces, it achieves 48% accuracy, well above the 1.05% random baseline, confirming that real fingerprinting signal exists, but far below the benchmark's 96.1%. More striking is what happens when we skip retraining: a model trained on the 2018 benchmark data and applied directly to our 2026 traces achieves just ∼1% accuracy. The reverse direction (training on our data, testing on the benchmark) also yields ∼1%. In other words, DF can learn to fingerprint websites *within* a dataset, but the patterns it learns are specific to the collection environment and time period. They do not transfer.

This has significant implications for the real-world threat of deep fingerprinting. An attacker who trains DF on traffic collected from their own network in one time period cannot expect the model to work on traffic from a different network or a later date without retraining. Maintaining an accurate model would require continuous, large-scale data collection matched to the target environment. This is a substantial operational burden that the headline 98% accuracy figure does not convey. Our results suggest that DF's practical threat may be meaningfully lower than benchmark evaluations imply, particularly against traffic collected under conditions that differ from training.

7 Limitations

Fixed threshold. Our open-world results use a single threshold ($\tau = 0.50$), which may understate performance by failing to capture the full operating curve. In practice, an attacker would tune τ to meet specific operational goals—lowering it to maximize recall or raising it to prioritize precision. A full threshold sweep, reporting ROC and precision–recall curves, would provide a more comprehensive performance profile.

WalkieTalkie open-world. Time constraints and the requirement for specialized half-duplex captures via a modified Tor Browser precluded an open-world evaluation of WalkieTalkie. Unlike other defenses, WalkieTalkie cannot be simulated on existing traces, creating a gap in our cross-defense comparison.

8 Subsequent and Future Work

The Deep Fingerprinting paper was published in 2018 and has since become one of the most cited works in the website fingerprinting literature. The years following its release have seen substantial activity on both the attack and defense sides.

Improved Attack Architectures While DF utilizes CNNs, the deep learning landscape has since shifted toward attention-based transformers. These models could theoretically outperform DF by capturing long-range packet dependencies that CNNs miss. However, it remains an open question whether these architectural gains can meaningfully bypass robust defenses like BuFLO or Tamaraw, or whether such defenses inherently neutralize any model architecture.

Adaptive Defenses None of the defenses evaluated in this study were designed or tuned with specific knowledge of the DF model's architecture, hyperparameters, or training procedure. Adaptive defenses that generate traffic perturbations specifically crafted to confuse a known classifier could perform substantially better, in line with the classic "cat and mouse game" that has come to represent the field of cybersecurity.

9 Conclusion

We reproduced the Deep Fingerprinting attack and evaluated it across six defense conditions, including two modern low-latency defenses not tested in the original paper. Our replication achieved 96.1% closed-world accuracy on undefended traffic, close to the original 98.3%; the small gap likely reflects minor differences in TensorFlow version and random initialization rather than any systematic implementation error. Against RegulaTor and BRO, accuracy dropped to 81.8% and 74.1%, meaningful reductions, but the attack remains well above chance. High-overhead defenses BuFLO and Tamaraw are more effective, pushing accuracy below 32%. These results suggest that low-overhead defenses, even those designed with deep learning attackers in mind, do not yet defeat the DF attack.

We identified an undocumented discrepancy: TCP packets vs. Tor cells, which affected all three independent collection efforts in our group. Correcting this on our 2026 data dropped accuracy to 48.0%, and cross-dataset experiments yielded random-chance accuracy (~1%) in both directions. This demonstrates that DF's high accuracy depends on distribution-specific patterns rather than universal fingerprints, and that any deployed DF model must be continuously retrained on fresh data. The practical threat of deep fingerprinting may therefore be lower than benchmark numbers suggest.

Acknowledgments

We want to acknowledge Akshay Srivatsan, David Mazières, Keith Winstein, and the CS244C teaching team at Stanford University for their guidance and insights, without which this paper would not be possible.

AI Usage

We made use of AI to help accelerate the development for this paper forward. Specifically, we used it for:

- **Paper Comprehension**: We used LLMs to deepen our understanding of the original Deep Fingerprinting paper and the broader subfield.
- **Code Development**: We used Claude Code to help with boilerplate and filler code.
- **Figure Generation**: We also used AI to help convert conceptual designs into figures.

References

[1] Xiang Cai, Rishab Nithyanand, Tao Wang, Rob Johnson, and Ian Goldberg. 2014. A Systematic Approach to Developing and Evaluating Website Fingerprinting Defenses. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14)*. ACM, 227–238. doi:10.1145/2660267.2660362

[2] Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium*. USENIX Association, San Diego, CA, USA, 303–320.

[3] Kevin P. Dyer, Scott E. Coull, Thomas Ristenpart, and Thomas Shrimpton. 2012. Peek-a-Boo, I Still See You: Why Efficient Traffic Analysis Countermeasures Fail. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy (SP '12)*. IEEE, San Francisco, CA, USA, 332–346. doi:10.1109/SP.2012.28

[4] Jamie Hayes and George Danezis. 2016. k-fingerprinting: A Robust Scalable Website Fingerprinting Technique. In *Proceedings of the 25th USENIX Security Symposium*. USENIX Association, Austin, TX, USA, 1187–1203.

[5] James K. Holland and Nicholas Hopper. 2022. RegulaTor: A Straightforward Website Fingerprinting Defense. *Proceedings on Privacy Enhancing Technologies* 2022, 2 (2022), 344–362. doi:10.56553/popets-2022-0049

[6] Marc Juarez, Mohsen Imani, Mike Perry, Claudia Diaz, and Matthew Wright. 2016. Toward an Efficient Website Fingerprinting Defense. In *Computer Security – ESORICS 2016 (Lecture Notes in Computer Science, Vol. 9878)*. Springer, Heraklion, Greece, 27–46. doi:10.1007/978-3-319-45744-4_2

[7] Colman McGuan, Chansu Yu, and Kyoungwon Suh. 2024. Practical and Lightweight Defense Against Website Fingerprinting. *Computer Communications* 228 (2024), 107976. doi:10.1016/j.comcom.2024.107976

[8] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. 2018. Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*. ACM, Toronto, ON, Canada, 1928–1943. doi:10.1145/3243734.3243768

[9] Tao Wang and Ian Goldberg. 2017. Walkie-Talkie: An Efficient Defense Against Passive Website Fingerprinting Attacks. In *Proceedings of the 26th USENIX Security Symposium*. USENIX Association, Vancouver, BC, Canada, 1375–1390.