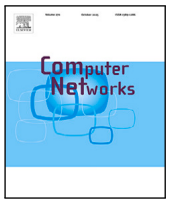
Computer Networks 292 (2027) 112937



Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Self-Training Mamba for source-free offline and online adaptation in encrypted traffic detection☆

Qi Liu, Hang Shen *, Tianjing Wang, Guangwei Bai

College of Computer and Information Engineering (College of Artificial Intelligence), Nanjing Tech University, 30 South Puzhu Road, Nanjing 211816, China

ARTICLE INFO

Keywords:
Encrypted traffic detection
Source-free domain adaptation
Self-Training Mamba
Distribution shift

ABSTRACT

Encrypted traffic detection models often experience substantial performance degradation in real-world deployments due to distribution shifts induced by geographical variability, VPN encapsulation, and protocol transformations. To address this issue, we propose Self-Training Mamba, a linear-time source-free domain adaptation framework that enables lightweight yet high-precision cross-domain encrypted traffic detection. The framework supports both offline adaptation for large-scale application traffic drift and online adaptation for rapidly evolving threats, sharing a unified adaptation module. A Mamba-based masked autoencoder is first pre-trained on source-domain traffic to learn domain-stable representations. During adaptation, a teacher–student self-training paradigm generates pseudo-labels for target-domain data. To mitigate pseudo-label noise without maintaining a historical memory bank, we introduce a reliability-aware filtering mechanism that selectively retains high-confidence samples. Uncertain samples are reconstructed to recover latent structural cues and enhance label quality, while a reliability ratio dynamically regulates teacher updates. We build a real-world dataset of encrypted traffic with geographically distributed receivers and heterogeneous access conditions to evaluate both adaptation modes. Extensive experiments on real-world and public benchmarks show that Self-Training Mamba significantly outperforms representative source-free baselines in cross-domain detection accuracy, while requiring fewer parameters and achieving faster inference.

1. Introduction

Real-world network environments have become increasingly heterogeneous due to the proliferation of network devices and emerging communication technologies [1]. Variations in infrastructure across geographical regions and access conditions produce distinct latency characteristics, congestion dynamics, and packet-level behaviors [2]. Dynamic routing, fragmented forwarding, and packet reassembly introduce further temporal irregularities and structural distortions in traffic sequences. On top of these effects, VPN and TLS encryption alter packet semantics and encapsulation patterns [3], masking informative features and inducing additional perturbations in the representation space.

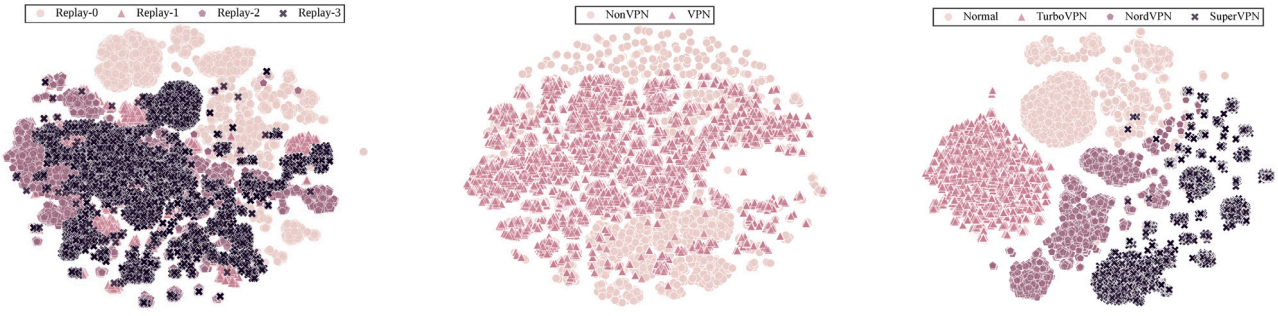
To qualitatively illustrate how domain shifts naturally arise in encrypted traffic, we present feature distributions from representative network conditions with increasing levels of discrepancy. Fig. 1(a) visualizes a real-world replay dataset collected across geographically distributed network environments with heterogeneous access conditions. The resulting feature distributions remain substantially overlapping but exhibit visible shifts, reflecting relatively light domain discrepancies induced by routing diversity, access variability, and cross-regional network conditions. Fig. 1(b) depicts a moderate shift in ISCXVPN2016 [4], where Non-VPN and VPN traffic exhibit increased inter-domain separation due to tunneling and encryption encapsulation. Fig. 1(c) illustrates a severe shift in AndroidVPN [5], where Normal traffic and traffic generated by different VPN implementations form clearly separated feature manifolds, reflecting substantial cross-protocol divergence.

These progressive patterns confirm that encrypted traffic in practical deployments exhibits multi-level distribution shifts rather than isolated anomalies. Such shifts arise from network infrastructure variability, protocol encapsulation, and heterogeneous access behaviors. Despite significant progress in encrypted traffic classification [6], including statistical feature-based approaches [7], deep sequential models [8,9], and graph-based architectures [10–12], most existing methods are designed and evaluated under the independent and identically distributed (i.i.d.) assumption. Although these models perform well

☆ This work was supported in part by the National Natural Science Foundation of China under Grants 61502230 and 61501224, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20201357, in part by the State Administration for Market Regulation Science and Technology Program under Grant 2025MK183, and in part by the Market Supervision Administration Science and Technology Fund of Jiangsu Province under Grants KJ2026027 and KJK2027027.

* Corresponding author.
E-mail address: hshen@njtech.edu.cn (H. Shen).

https://doi.org/10.1016/j.comnet.2026.112937
Received 11 August 2026; Received in revised form 11 October 2026; Accepted 7 November 2026
Available online 15 November 2026
1389-1286/© 2026 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.



(a) Real-world replay datasets. (b) ISCXVPN2016. (c) AndroidVPN.

Fig. 1. Visualization of domain distributions under different levels of domain shift. (a) Feature clusters exhibit strong overlap, indicating high inter-domain similarity and light drift. (b) Partial inter-domain separation becomes evident, reflecting a moderate shift. (c) Feature clusters are well separated with clear boundaries, corresponding to a severe shift.

in stable environments, this assumption does not hold in real-world multi-level shifts, leading to brittle, unpredictable performance in cross-domain scenarios [13]. These observations underscore the need for adaptive mechanisms capable of operating under inherently non-i.i.d. conditions.

Domain adaptation (DA) has been widely adopted as a transfer learning paradigm that allows models trained on a source domain to generalize to a target domain with limited or no labels [14,15]. Traditional DA relies on feature alignment or distribution matching and implicitly assumes sufficient structural similarity between the source and target domains. However, under moderate or severe shifts, as illustrated in Fig. 1(b) and (c), the weakened cross-domain correspondence makes source-to-target transfer and distribution alignment substantially more difficult. Consequently, a source-trained model may exhibit limited direct generalization to target traffic, requiring effective target-domain adaptation. Moreover, access to historical source-domain traffic may be restricted by storage, privacy concerns, and runtime overhead [16], limiting the feasibility of DA for enterprise-scale or latency-sensitive encrypted traffic detection.

Source-free domain adaptation (SFDA) [17,18] relaxes these constraints by updating only the deployed model without access to source data. Current SFDA methods can be broadly categorized into data-centric approaches that approximate source-like distributions [19,20] and model-centric approaches that selectively update model components [21]. Although many existing SFDA methods are developed for static or mildly shifted domains [22], recent studies have extended adaptation to dynamic target distributions and open-set scenarios, reflecting growing attention to evolving target characteristics [23]. In encrypted traffic, such evolution is further compounded by spatiotemporal variability, protocol heterogeneity, and changing usage behaviors, resulting in more complex multi-level distribution shifts and placing higher demands on adaptation robustness and responsiveness.

1.1. Challenging issues and related works

Although DA techniques offer promising solutions for handling cross-domain encrypted traffic, the multi-level shifts in real deployments introduce several challenges.

(1) Limited source-model generalization under severe cross-protocol domain shifts. Traffic evolution in real network environments often exhibits complex distribution shifts at the protocol layer, particularly strong heterogeneous shifts induced by different VPN protocols across various layers. In real-world scenarios, access to historical source-domain traffic may be restricted, motivating adaptation from a pre-trained source model without retaining source data. However, severe cross-protocol discrepancies limit the source model's direct generalization, while adaptation based solely on unlabeled target data lacks explicit target-domain supervision to correct target-specific decision boundaries. Typical semi-supervised DA (SSDA) methods, such as MME [24] and CDAC [25], introduce a small amount of labeled target data as supervisory guidance, providing explicit target-domain information and facilitating more reliable class boundaries [26]. Furthermore, SHOT++ [27] extends SFDA to source-free SSDA (SF-SSDA) through hypothesis transfer to improve adaptability. To advance SF-SSDA, PIDM [28] designs a progressive intra-domain mixing strategy and a multi-scale pseudo-label refinement mechanism to alleviate instability in representation optimization. The detection networks in the aforementioned SF-SSDA methods rely on ResNet [29] or VGG [30], creating bottlenecks in inference efficiency under real-world network conditions.

(2) Pseudo-label noise accumulation without historical memory banks. Mean Teacher [31], based on consistency regularization, has been widely adopted in SFDA [32] and SSDA [33] for pseudo-label generation. However, pseudo-label noise remains a critical issue under domain shifts. Existing methods, such as G-SFDA [20] and SHOT [34], refine pseudo-labels through memory banks, whereas curriculum-based selection [35] reduces storage requirements by selecting reliable samples. Recent studies further exploit calibration and uncertainty estimation to improve prediction reliability, including Cal-SFDA [36] and UBD [37]. Related efforts in concealed object segmentation and unpaired image enhancement also demonstrate the effectiveness of leveraging ambiguous observations rather than discarding them [38–40]. However, fixed exponential moving average (EMA) updates adopted by many self-training frameworks may introduce temporal lag under rapidly evolving traffic distributions, aggravating pseudo-label noise accumulation.

(3) Real-time adaptation demands under rapidly evolving encrypted traffic. Real-world encrypted traffic detection requires diverse adaptations. Real-time monitoring requires detection systems to adapt immediately to rapidly evolving threat patterns, while large-scale application traffic distribution shifts require sufficient offline retraining to ensure robust performance. Network traffic DA methods primarily focus on offline batch processing [41], but struggle to meet the real-time adaptation requirements for streaming traffic. Online test-time adaptation (OTTA) [42] enables continuous, real-time optimization of model parameters as data streams arrive, providing an efficient adaptation strategy suitable for network security. However, some image processing methods (such as NC-TTT [43] and TTT-MAE [44]) require multiple rounds of gradient optimization during the testing phase to adapt to target domains. Facing real-time threats from attack traffic, detection models must achieve satisfactory accuracy with a single iteration, placing stringent requirements on lightweight design and robust adaptation.

1.2. Contribution and organization

Effectively tackling these challenges requires an adaptation mechanism that produces domain-stable representations, mitigates pseudo-label accumulation, and supports responsive updates under streaming encrypted traffic. To this end, we develop a Mamba-based self-training DA framework operating on a structured network flow matrix (NFM). The NFM representation preserves hierarchical protocol semantics and temporal dependencies, while Mamba's selective state-space [45] formulation enables linear-time sequence modeling and implicit regularization against domain-induced distortions. The framework supports both offline robust adaptation and lightweight online adaptation, enabling accurate and resource-efficient cross-domain encrypted traffic detection. The main contributions are summarized as follows:

- *Efficient Mamba-based cross-domain detection:* A unified offline–online self-training framework is developed under the SF-SSDA paradigm. By integrating Mamba with the structured NFM representation, the framework captures long-range dependencies and protocol-level characteristics in encrypted traffic, providing an efficient representation basis for adaptation across heterogeneous network and protocol shifts.
- *Reliability-aware pseudo-label optimization:* A memory-bank-free filtering approach is introduced to selectively retain high-confidence pseudo-labels and suppress unreliable ones. A Mamba-based decoder reconstructs samples with uncertain labels to recover latent structures and refine pseudo-label quality. A reliability ratio is incorporated into the update momentum to stabilize teacher–student learning and accelerate convergence.
- *Instant adaptation against real-time threat:* The offline adaptation mechanism is extended to a lightweight online mode that continuously adapts to streaming encrypted traffic with bounded memory overhead. This enables real-time model updates without retraining or retaining source data, supporting rapid responses to evolving behaviors.

To validate the effectiveness and practicality of the proposed framework, we address the following questions through systematic experimentation:

- **RQ1:** Across heterogeneous domain shifts, can the proposed method achieve accurate cross-domain detection while maintaining low inference overhead?
- **RQ2:** How does the reliability-aware pseudo-label filtering mechanism affect the convergence and stability of offline adaptation?
- **RQ3:** Can the online adaptation scheme effectively adapt to continuously evolving distribution shifts in streaming encrypted traffic?

The remainder of this paper is organized as follows. Section 2 introduces the offline adaptation framework with a Mamba-based architecture and pseudo-label optimization. Section 3 extends the framework to online adaptation for streaming encrypted traffic. Section 4 summarizes the experimental environments, datasets, and implementation details. Performance evaluation and analytical results are presented in Section 5. Finally, Section 6 concludes the paper and outlines ongoing research.

2. Offline adaptation method

To support robust cross-domain detection, we develop a universal offline adaptation framework under the SF-SSDA paradigm. The overall process consists of two components: source-domain pre-training using masked autoencoders (MAE) and target-domain adaptation without access to source data. Let $\mathcal{D}_s = \{(x, y) \sim u_s(x, y)\}$ denote the source domain dataset, where x represents network traffic samples, u_s denotes the source domain distribution, and $y \in \mathcal{Y}$ is the traffic category. The corresponding target domain dataset is represented as $\mathcal{D}_t = \{(x, y) \sim u_t(x, y)\}$, belonging to network environments with distribution differences from the source domain (such as different protocols, traffic patterns, or network types). Although $\mathcal{D}_s$ and $\mathcal{D}_t$ share the same semantic label space $\mathcal{Y}$, their feature distributions differ significantly, violating the i.i.d. assumption and necessitating a model $f : x \to y$ that remains stable across varying degrees of domain shift.

2.1. Domain-agnostic encoder-decoder

The NFM representation transforms raw traffic into a structured matrix, which is subsequently partitioned into a sequence of discrete patches for processing. Modeling such patch sequences requires capturing dependencies across distant spatial regions of the NFM while maintaining computational efficiency. Transformer-based architectures [46] incur quadratic complexity, limiting deployment under latency constraints, while recurrent models suffer from vanishing gradients in long sequences. Accordingly, we adopt Mamba as the encoder–decoder backbone for its linear complexity and selective state-space formulation. State space models (SSMs) are derived from continuous-time dynamical systems. To enable processing of discrete sequences, the continuous formulation is discretized via zero-order hold (ZOH) as

$$h_t = \overline{\mathbf{A}} h_{t-1} + \overline{\mathbf{B}} x_t, \quad y_t = \mathbf{C} h_t, \tag{1}$$

with $\overline{\mathbf{A}} = \exp(\Delta\mathbf{A})$ and $\overline{\mathbf{B}} \approx \Delta\mathbf{B}$, with Δ denoting the discretization step. Mamba extends SSMs by introducing a selective state mechanism in which $\mathbf{B}$, $\mathbf{C}$, and Δ are dynamically modulated as input-dependent functions, enabling content-aware processing that amplifies discriminative traffic patterns while suppressing domain-induced noise.

In contrast to handcrafted positional encodings or rigid packet tokenization, the state-space formulation provides a flexible inductive bias for encrypted traffic. By leveraging Mamba's adaptive filtering and linear scalability, the encoder extracts stable temporal representations that are less sensitive to domain-dependent structural variations. This selective mechanism implicitly regularizes representation drift by emphasizing invariant temporal cues and down-weighting noise induced by routing conditions, protocol encapsulation, or heterogeneous access environments.

The encoder stacks Mamba blocks to extract hierarchical representations from NFM patches. A lightweight Mamba decoder reconstructs masked patches during pre-training and processes unreliable samples during adaptation. Encoded features are aggregated via global average pooling and projected to the category space through a linear classifier.

2.2. Source domain pre-training

In the source domain, the MAE is employed to reconstruct the original NFM, enabling the learning of essential feature representations of encrypted traffic without over-relying on label information. The MAE consists of an encoder and a decoder. The NFM undergoes partitioning into K fixed-size patches, where the reconstructed and original values of the kth patch are denoted as x'_k and x_k, respectively. A high proportion of patches are randomly masked using a binary mask, $m_k \in \{0, 1\}$ (1 for masked, 0 for retained). The encoder only maps unmasked patches into latent representations, while the lightweight decoder reconstructs the complete NFM from mask tokens and encoded features. The reconstruction loss is formulated as

$$\mathcal{L}_1 = \frac{\sum_{k=1}^{K} \left[\left(x'_k - x_k \right)^2 \cdot m_k \right]}{\sum_{k=1}^{K} m_k}. \tag{2}$$

This masking compels the model to capture global dependencies rather than focusing only on local patterns. For downstream traffic classification tasks, the encoder parameters are loaded from the MAE-based self-supervised model, and the decoder is replaced with a task classification head. Encoded features undergo statistical standardization through

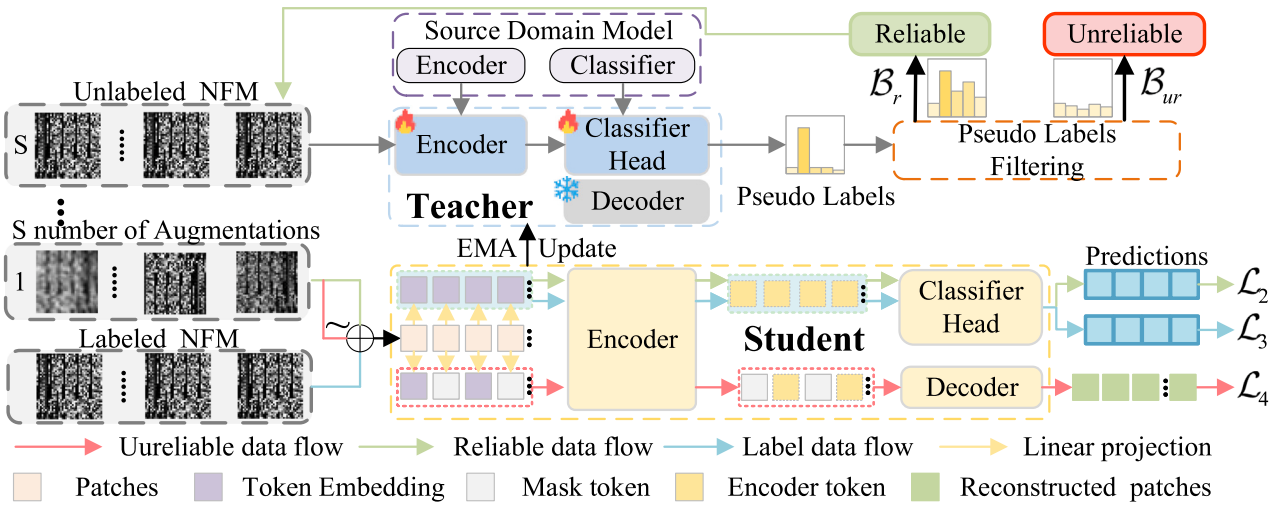


Fig. 2. Mamba-based self-training framework for offline domain adaptation.

batch normalization layers, and are subsequently linearly projected to C-dimensional category space. Let $\hat{y}_c$ and y_c represent the predicted probability and true label for category c. The loss for the classification task follows cross-entropy

$$\mathcal{L}_{CE} = -\sum_{c=1}^{C} y_c \log\left(\hat{y}_c\right). \tag{3}$$

2.3. Target domain training

As shown in Fig. 2, the encoder and classifier of the teacher (f_{θ_t}) and student models (f_{θ_s}) are initialized with parameters inherited from the source-domain model, providing a source-informed initialization for subsequent target-domain adaptation. The teacher model focuses on pseudo-label generation, with the decoder frozen to avoid unnecessary reconstruction computation; the student model's decoder adopts a self-supervised training strategy to reconstruct unreliable samples to mine feature representations.

Consider a training batch containing a labeled sample set $\mathcal{B}_l \subset \mathcal{D}_t$ and an unlabeled sample set $\mathcal{B}_u \subset \mathcal{D}_t$. For $(x_j, y_j) \in \mathcal{B}_l$, each input x_j is first segmented into fixed-size patches and converted into token embeddings, which are then fed into the student encoder and classifier to obtain the prediction $f_{\theta_s}(x_j)$. The supervised loss is formulated as

$$\mathcal{L}_2 = \frac{1}{|\mathcal{B}_l|} \sum_{(x_j,y_j)\in\mathcal{B}_l} \mathcal{L}_{CE}(f_{\theta_s}(x_j), y_j) \tag{4}$$

where $\mathcal{L}_{CE}$ represents the cross-entropy loss. For unlabeled NFM, denoted as $x_i \in \mathcal{B}_u$, the student model independently predicts the result as $f_{\theta_s}(x_i)$; while the teacher model first applies mixed augmentation, then extracts features through the encoder and generates initial pseudo-labels $\hat{y}_i$ through the classifier. Given that low-quality pseudo-labels lead to negative propagation, a memory-bank-free pseudo-label filtering mechanism is designed to enhance label reliability and diversity. Pseudo-label reliability is measured by a binary score $R_i \in \{0,1\}$ and the secondary screening indicator $A_i \in \{0,1\}$. Accordingly, $\mathcal{B}_u$ is divided into unreliable and reliable subsets, $\mathcal{B}_{ur}$ and $\mathcal{B}_r$ (detailed in Section 2.5). For $x_i \in \mathcal{B}_r$, the teacher–student collaborative learning loss is calculated as

$$\mathcal{L}_3 = \frac{1}{|\mathcal{B}_r|} \sum_{x_i\in\mathcal{B}_r} \mathcal{L}_{CE}(f_{\theta_s}(x_i), \hat{y}_i). \tag{5}$$

For $x_i \in \mathcal{B}_{ur}$, we employ self-supervised reconstruction to learn universal traffic representations (similar to source domain pre-training). The reconstruction loss is given by

$$\mathcal{L}_4 = \frac{1}{|\mathcal{B}_{ur}|} \sum_{x_i\in\mathcal{B}_{ur}} \|x_i - x_i'\|_2^2 \tag{6}$$

where x_i' denotes the reconstructed representation of $x_i \in \mathcal{B}_{ur}$. This design enables the model to continuously learn essential traffic feature representations and promote the migration of pseudo-labels from unreliable to reliable. The overall loss is quantified as a weighted combination of (4), (5), and (6), i.e.,

$$\mathcal{L} = \mathcal{L}_2 + \beta_1\mathcal{L}_3 + \beta_2\mathcal{L}_4 \tag{7}$$

where β_1 and β_2 are weighting factors to balance reconstruction loss and collaborative learning loss.

2.4. Pseudo-label generation

To provide reliable pseudo-labels for unlabeled samples, a teacher–student model framework based on self-training is adopted. Unlike traditional pre-training teacher models, our teacher model is structured as an EMA of the student model. This update mechanism facilitates knowledge acquisition from unlabeled encrypted traffic. Formally, the teacher and student models share the same architecture but use different parameter update strategies during training. Let $\theta_s^{(e)}$ and $\theta_t^{(e)}$ represent the student and teacher model parameters at iteration e, respectively. Initially, $\theta_s^{(0)} = \theta_t^{(0)}$. At the eth iteration, the teacher model parameters are updated by

$$\theta_t^{(e)} = \lambda\theta_t^{(e-1)} + (1-\lambda)\theta_s^{(e)} \tag{8}$$

where $\lambda \in [0,1]$ is the momentum coefficient that adjusts the teacher's adaptation speed to student characteristics at each iteration. The student model is updated by minimizing (7). Noise in pseudo-labels is inevitable during self-training, leading to error accumulation and unstable model training. Pseudo-label filtering mechanisms are commonly introduced to improve pseudo-label reliability.

2.5. Memory-bank-free pseudo-label filtering

To avoid the storage overhead and complexity introduced by memory-bank-based pseudo-label filtering, we design a memory-bank-free mechanism that does not maintain historical target samples or predictions across batches. In general, model confidence correlates with accuracy in well-calibrated models [47], while uncertainty reflects prediction reliability [48]. To assess pseudo-label reliability, we integrate multi-view confidence and uncertainty to construct a unified reliability metric for filtering.

Fig. 3 provides an overview of the proposed pseudo-label filtering process. For $x_i \in \mathcal{B}_u$, N complementary views are generated through a mixed augmentation sequence S, denoted as $\tilde{x}_{i,n} = S_n(x_i)$ for $n \in [1, N]$.

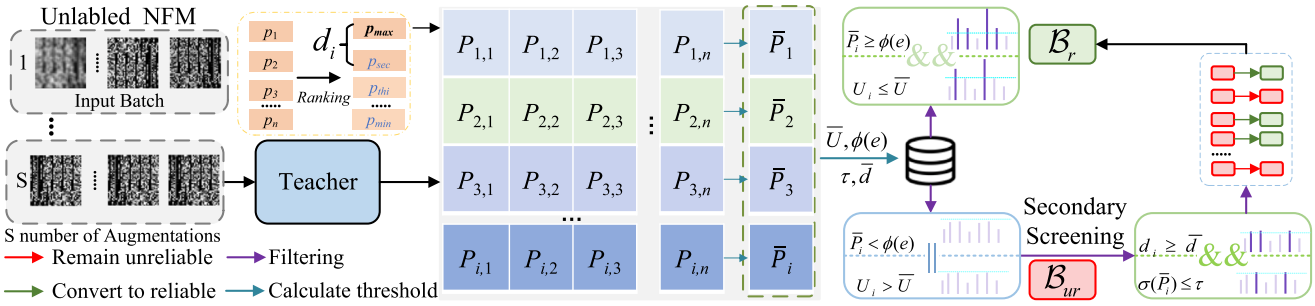


Fig. 3. Overview of the memory-bank-free pseudo-label filtering mechanism.

The teacher model f_{θ_t} produces prediction logits for each augmented view as

$$z_{i,n} = f_{\theta_t}(\tilde{x}_{i,n}). \tag{9}$$

The corresponding class probability distribution is

$$p_{i,n} = \mathrm{softmax}(z_{i,n}). \tag{10}$$

To obtain a robust pseudo-label, the class probabilities from different views are aggregated as

$$\bar{p}_i = \frac{1}{N}\sum_{n=1}^{N} p_{i,n}. \tag{11}$$

The pseudo-label is then assigned according to the class with the highest aggregated probability:

$$\hat{y}_i = \arg\max_{c\in\{1,\dots,C\}} \bar{p}_{i,c}. \tag{12}$$

Meanwhile, the confidence of each augmented view is calculated as $P_{i,n} = \max_c(p_{i,n,c})$. The mean of multi-view confidence is calculated as $\bar{P}_i = \frac{1}{N}\sum_{n=1}^{N} P_{i,n}$. The batch-level mean confidence is further computed as $\bar{P} = \frac{1}{|\mathcal{B}_u|}\sum_{i=1}^{|\mathcal{B}_u|} \bar{P}_i$. Consequently, in the eth ($\leq E$) training iteration, the confidence threshold is computed as

$$\phi(e) = \bar{P}(1 - \alpha(e/E)) \tag{13}$$

where α is a weight factor. We maintain a higher threshold during early training to ensure pseudo-label quality, then relax it moderately in later stages to introduce more diverse pseudo-labels.

Although multi-view aggregation improves pseudo-label robustness, prediction reliability may still vary under domain shifts. Therefore, we further characterize prediction uncertainty from the perspectives of cross-view stability and class-level ambiguity. The dispersion of the multi-view confidence scores is calculated as

$$\sigma_i = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(P_{i,n} - \bar{P}_i\right)^2}, \tag{14}$$

which characterizes the variation in prediction confidence across augmented views and reflects cross-view prediction stability. To further characterize class-level uncertainty in multi-class predictions, we aggregate the class probabilities obtained from different augmented views. The aggregated probability for class $c \in \{1, \dots, C\}$ is defined as

$$Q_{i,c} = \frac{1}{N}\sum_{n=1}^{N} p_{i,n,c}. \tag{15}$$

Based on the aggregated class distribution, the predictive entropy is normalized as

$$\mathcal{H}_i = -\frac{\sum_{c=1}^{C} Q_{i,c}\log Q_{i,c}}{\log C}. \tag{16}$$

A larger $\mathcal{H}_i$ indicates greater ambiguity in the aggregated class distribution. Together, σ_i and $\mathcal{H}_i$ capture complementary aspects of prediction uncertainty, namely cross-view stability and class-level ambiguity. The overall uncertainty score is determined as

$$U_i = \alpha\sigma_i + (1-\alpha)\mathcal{H}_i, \tag{17}$$

where $\alpha \in [0,1]$ balances cross-view dispersion and entropy-based uncertainty.

Combining U_i with the adaptive confidence threshold $\phi(e)$, the reliability of x_i is determined as

$$R_i = \begin{cases} 1, & \text{if } \bar{P}_i \geq \phi(e) \text{ and } U_i \leq \bar{U}, \\ 0, & \text{otherwise}, \end{cases} \tag{18}$$

where $\bar{U}$ denotes the mean uncertainty over the current unlabeled batch. The primary filtering criterion may preferentially retain samples from dominant categories. We therefore introduce the difference of confidence (DoC) between the top-1 and top-2 predictions as a secondary screening criterion. Let $\mathcal{B}'_u = \{x_i \in \mathcal{B}_u \mid R_i = 0\}$ denote the subset rejected by the primary filtering stage. For $x_i \in \mathcal{B}'_u$, let d_i denote its DoC value. The secondary screening criterion is defined as

$$A_i = \begin{cases} 1, & \text{if } d_i \geq \bar{d} \text{ and } \sigma_i \leq \tau, \\ 0, & \text{otherwise}, \end{cases} \tag{19}$$

where $\bar{d}$ and τ denote the mean DoC and mean cross-view dispersion, respectively, over $\mathcal{B}'_u$. A large d_i indicates a clear separation between the most probable class and its closest competitor, whereas a small σ_i indicates stable predictions across augmented views. Combining R_i and A_i therefore retains reliable pseudo-labels while recovering additional samples with well-separated and stable predictions.

Traditional EMA model update methods adopt fixed momentum coefficients, which are difficult to adapt to time-varying traffic. Given the differentiability of pseudo-labels, we use the reliability ratio to adjust the momentum coefficient. The unlabeled sample set is divided into reliable and unreliable subsets $\mathcal{B}_r$ and $\mathcal{B}_{ur}$.

$$\begin{cases} \mathcal{B}_r = \{x_i | R_i = 1 \cup A_i = 1\} \\ \mathcal{B}_{ur} = \{x_i | R_i = 0 \cap A_i = 0\}. \end{cases} \tag{20}$$

Note that they are conceptually disjoint sets rather than an actual memory bank. Let λ_e represent the eth iteration's momentum coefficient, where this iteration's reliability ratio is calculated as $\rho_e = |\mathcal{B}_r^{(e)}|/(|\mathcal{B}_r^{(e)}| + |\mathcal{B}_{ur}^{(e)}|)$, with $\rho_e \in [0,1]$. The momentum coefficient is then updated based on (8) as

$$\lambda_e = \gamma\lambda_{e-1} + (1-\gamma)(1-\rho_e) \tag{21}$$

where $\gamma \in [0,1]$ controls the temporal smoothing of the momentum coefficient. A higher reliability ratio reduces λ_e, assigning greater weight to the current student model and accelerating knowledge transfer, whereas a lower reliability ratio increases λ_e, allowing the teacher to retain more historical information and maintain update stability.

3. Online adaptation method

The offline method introduced earlier enables robust adaptation to encrypted application traffic shifts under the SF-SSDA paradigm. Nevertheless, real-time intrusion detection systems must operate on streaming traffic without access to ground-truth labels, necessitating immediate and continuous adaptation to evolving threats. To satisfy this demand, we propose an online extension of our framework based on OTTA.

In contrast to the offline setting, where a small number of labeled target samples enable semi-supervised learning, the online scenario assumes a fully unsupervised setup. For each incoming sample x_q in the sequential network traffic stream $\{x_1, x_2, \ldots, x_Q\}$, the offline objective in (7) is reformulated on a per-sample basis according to the reliability of its pseudo-label $\hat{y}_q$. Specifically, the online reliability criterion follows the offline filtering strategy, where the primary confidence-based indicator and the secondary DoC-based screening are jointly considered. The final reliability indicator is defined as $\tilde{R}_q = R_q \vee A_q$. Cross-entropy supervision is applied when the pseudo-label is reliable, whereas an MSE reconstruction loss is used otherwise. The resulting sample-level online loss is

$$\mathcal{L}_q = \tilde{R}_q \cdot \mathcal{L}_{CE}(f_{\theta_s}(x_q), \hat{y}_q) + (1 - \tilde{R}_q) \cdot \left\| x_q - x'_q \right\|_2^2. \tag{22}$$

Since batch data cannot be obtained at once, the pseudo-label filtering strategy requires special improvement. Considering the phenomenon of traffic delays or bursts in network environments, traditional fixed-time window methods may cause many samples to flood in during network traffic peaks, leading to dramatic increases in memory consumption or even overflow. In contrast, during low-traffic periods, the number of samples within the window is sparse, making it unable to provide sufficient statistical information. To address this, we introduce a buffer queue for pseudo-label filtering in the online method.

For each newly arrived sample x_q, the same multi-view augmentation strategy as in offline adaptation is applied to obtain multiple predictions. The corresponding reliability statistics, including the mean confidence $\bar{P}_q$, uncertainty score U_q, cross-view dispersion σ_q, and DoC value d_q, are calculated and stored in a lightweight queue. Because the queue-based thresholds require sufficient recent observations for stable estimation, the online procedure begins with a short initialization stage. During this stage, the reliability statistics are accumulated in the queue, while pseudo-label refinement and model updates remain inactive. Once the queue accumulates L entries, it retains the reliability statistics of the most recent L samples, $\left\{(\bar{P}_j, U_j, \sigma_j, d_j)\right\}_{j=q-L+1}^{q}$. The mean confidence of the current queue is calculated as

$$\bar{P}'_q = \frac{1}{L} \sum_{j=q-L+1}^{q} \bar{P}_j. \tag{23}$$

The online confidence threshold is then defined as

$$\phi_q = (1 - \alpha)\bar{P}'_q. \tag{24}$$

For each subsequent sample, the newly calculated statistics are appended to the queue and the oldest entry is removed. Therefore, $\bar{P}'_q$ and the corresponding confidence threshold are continuously updated according to the recent traffic statistics.

The mean uncertainty $\bar{U}_q$ used by the primary screening criterion is estimated over all entries in the current queue. After the primary screening step, the mean DoC value $\bar{d}_q$ and the cross-view dispersion threshold τ_q used by the secondary screening criterion are calculated from the entries rejected by the primary criterion, consistent with the rejected subset in the offline formulation. The reliability indicators R_q and A_q otherwise follow the criteria in (18)–(19), with the batch-level statistics replaced by the corresponding queue-based statistics. Moreover, λ in (21) is adjusted according to the reliability ratio calculated from the reliable and unreliable samples identified within the queue, enabling consistent teacher updates across both adaptation modes.

Table 1
Real-world traffic replay configuration.

Env. ID	Sender Loc.	Receiver Loc.	Access
Replay-0	China	China	Ethernet
Replay-1	China	LA, USA	Ethernet
Replay-2	China	Singapore	Wi-Fi
Replay-3	China	London, UK	5G

After evaluating the pseudo-label reliability of the samples in the current mini-batch, the sample-level losses in (22) are averaged to perform one model update. Therefore, the queue continuously maintains recent traffic statistics and provides local distribution information for threshold adaptation, rather than relying on the entire historical stream. Compared with memory bank methods that require storing $O(N)$ historical prediction information, the queue introduces only $O(L)$ memory overhead, where L is the fixed queue length, independent of the total stream length. Furthermore, the OTTA method avoids full retraining when encountering new shifts. Instead, it performs lightweight parameter updates online during the detection process. The overall computational cost consists of inference latency and adaptation time; the latter refers to the time spent updating the model online.

4. Experiment preparation

4.1. Dataset preparation and analysis

We constructed a real-world evaluation dataset by replaying TLS-encrypted traffic from the CIRA-CIC-DoHBrw-2020 dataset[1] between a sender in China and geographically distributed receivers located in China, the United States, Singapore, and the United Kingdom. The replay environments employ heterogeneous access technologies, including Ethernet, Wi-Fi, and 5G, as summarized in Table 1. The resulting dataset contains 15,000 malicious and 15,000 benign TLS flows. Replay-0 is designated as the source domain, while Replay-1, Replay-2, and Replay-3 serve as the target domains. By varying geographical paths, access technologies, and network conditions while preserving the underlying traffic semantics, these replay environments introduce environment-induced distribution shifts representative of cross-network deployment scenarios.

To prevent paired-flow leakage across replay environments, data partitioning was performed according to the identities of the original flows. Specifically, all replayed instances derived from the same original flow were assigned to the same data split. Therefore, the sets of original flow identities used for source-domain training, target-domain adaptation, and target-domain testing are mutually exclusive. A target test flow, as well as any replayed counterpart originating from the same original flow, is thus never observed during either source-domain training or target-domain adaptation. This identity-level partitioning prevents the model from exploiting duplicated or paired traffic content across replay environments and ensures that the reported performance reflects cross-domain adaptation rather than memorization of previously observed flows.

We further evaluate the proposed method on two public encrypted-traffic benchmarks. ISCXVPN2016 [4] characterizes the distribution shift introduced by VPN tunneling and encryption. Non-VPN traffic is used as the source domain and VPN traffic as the target domain, with six shared application categories: Browsing, Email, Chat, Streaming, File Transfer, and VoIP. AndroidVPN [5] provides a more challenging cross-protocol setting. Normal traffic is used as the source domain, while traffic generated through NordVPN, TurboVPN, and SuperVPN constitute three target domains. These VPN implementations employ

[1] https://www.unb.ca/cic/datasets/dohbrw-2020.html

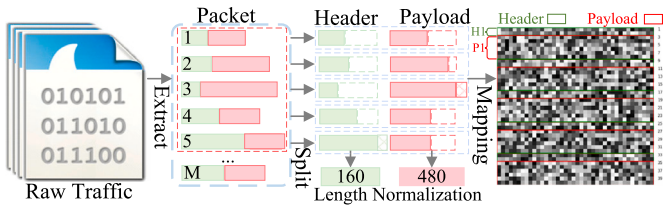


Fig. 4. Network flow information matrix construction.

WireGuard [49], IKEv2/IPsec,[2] and PPTP,[3] respectively. The dataset contains UDP-based VPN traffic covering 150 categories, with 6000 training flows and 1500 testing flows. As visualized in Fig. 1, the real-world replay dataset, ISCXVPN2016, and AndroidVPN collectively cover progressively more challenging domain shifts, ranging from relatively light environment-induced variations to moderate VPN-induced shifts and severe cross-protocol discrepancies.

4.2. Data preprocessing and traffic representation

Data preprocessing follows the hierarchical flow representation scheme adopted in recent traffic analysis [50,51] and consists of traffic segmentation and representation construction. We used SplitCap[4] to segment the raw network traffic into individual flows. The resulting pcap files were organized according to the application-category identifiers provided by the original dataset. Subsequently, the first M IP packets (including IPv4 and IPv6) from each flow were extracted in chronological order, where $M = 5$.

Before constructing the representation shown in Fig. 4, fields that could explicitly reveal endpoint or environment identities were sanitized. Specifically, source and destination IP addresses were masked in the packet headers, while packet-capture timestamps and other pcap-level metadata were excluded from the model input. The remaining protocol-related header fields and payload bytes were then converted into hexadecimal representations.

To handle variable packet lengths, a fixed-length representation was adopted. For each packet, the header and payload were standardized to 160 hexadecimal characters (80 bytes) and 480 hexadecimal characters (240 bytes), respectively. Sequences exceeding these limits were truncated, whereas shorter sequences were zero-padded to the predefined lengths. Consequently, each flow contains 3200 hexadecimal characters, i.e., $5 \times (160 + 480)$, corresponding to 1600 bytes. Each byte was mapped to one grayscale value, yielding 1600 pixel values that were subsequently reshaped into a 40×40 NFM.

As illustrated in Fig. 4, this mapping preserves the packet order: rows 1–8 correspond to the first packet, rows 9–16 to the second packet, and so forth. Within each 8-row block, the first two rows encode header information and the remaining six rows encode payload information. The resulting NFM therefore preserves packet-level temporal ordering and protocol structure while providing a fixed-size representation for subsequent feature learning.

4.3. Comparison methods

To validate the effectiveness of the offline adaptation approach (Proposed-1), we selected several representative SSDA baselines. S+T is a simple transfer method that jointly trains on source-domain data and few-shot labeled target samples. MME [24] enhances target-domain adaptation by maximizing the marginal entropy of target-domain samples near classification boundaries. CDAC [25] achieves cross-domain feature alignment through adversarial clustering and confidence-weighted pseudo-labels. SHOT++ [27] achieves efficient DA by combining supervised training on labeled target domain data with self-supervised learning on unlabeled data, without accessing source domain data.

Additionally, to disentangle the effects of the backbone architecture and adaptation strategy, we construct three matched-backbone Mamba variants: Mamba-S, Mamba-MT, and Mamba-SHOT++. Mamba-S denotes the source-trained Mamba model directly evaluated on the target domains without using any target-domain data. Mamba-MT and Mamba-SHOT++ follow the Mean Teacher and SHOT++ adaptation paradigms, respectively, using the same labeled and unlabeled target splits as Proposed-1. All variants adopt the same Mamba backbone and source-domain MAE-pretrained initialization as Proposed-1. These controls quantify direct source-model generalization and enable a fair comparison of different adaptation strategies under a unified backbone and pre-training scheme.

To comprehensively evaluate each method in SSDA-related experiments, each target-domain category contains 1, 5, 10, and 20 labeled samples. For each few-shot setting, we conduct five independent runs with different labeled-sample selections and random seeds, and report the mean accuracy with standard deviation.

For real-time adaptation evaluation, we compare the online approach (Proposed-2) with AdaContrast [52], which employs a compact memory queue for online contrastive learning and real-time pseudo-label refinement to handle streaming data. In addition, we replace the backbone of Proposed-2 (Mamba) with a Transformer (Proposed-3) to investigate architectural scalability and the impact of sequence modeling approaches on online adaptation performance.

To evaluate the robustness of online adaptation, all online experiments are repeated over five independent runs. For each run, an independent target-stream realization is generated using a different random seed, while the same stream realization is shared by all compared methods to ensure a paired and fair comparison. All methods start from their corresponding source-trained checkpoints, and each target flow is observed only once during adaptation. No target labels are used for model updating. The stochastic data augmentation and online optimization processes are independently initialized in each run. We report the mean accuracy and standard deviation over the five runs.

4.4. Implementation details

During source-domain training, the batch size was set to 128 with 150,000 training steps. AdamW learning rate was initialized to 1.0×10^{-3} with a linear scaling strategy. The random masking ratio was set to 0.9. During offline target-domain adaptation, all models were optimized for 6000 iterations. The labeled and unlabeled target samples were sampled separately. The unlabeled mini-batch size was set to 64, while the labeled batch size was set to $\min(64, |D_l|)$. For all online experiments, the incoming mini-batch size was set to 16.

α in (13), (17) and (24) were set to 0.1, while γ in (21) was set to 0.6. β_1 and β_2 in (7) were set to 0.2 and 0.6. The momentum coefficient was initialized to 0.8. Proposed-2 and Proposed-3 encoders and decoders consist of 4 Mamba (Transformer) blocks and 2 Mamba (Transformer) blocks, respectively. All SSDA baseline methods and AdaContrast used the ResNet34 backbone. Our method was implemented in PyTorch 2.1.1, with experiments conducted on Ubuntu 22.04, equipped with an Intel Core i9-13900 K and two NVIDIA RTX 4090 24 GB GPUs.

5. Experimental results

The following experiments evaluate the proposed framework with respect to the three research questions (**RQ1–RQ3**) introduced above. Section 5.1 evaluates offline adaptation across domain shifts and analyzes convergence and component effectiveness (**RQ1** and **RQ2**). Section 5.2 evaluates online adaptation under streaming conditions through detection performance, ablation, and queue-length sensitivity (**RQ3**). Finally, Section 5.3 examines the computational efficiency of offline and online adaptation, complementing **RQ1** and **RQ3**.

[2] https://www.rfc-editor.org/rfc/rfc7296
[3] https://www.rfc-editor.org/rfc/rfc1548
[4] https://www.netresec.com/?page=SplitCap

Table 2
Accuracy (%) on ISCXVPN2016.

Method	Params	Source-Free	NonVPN→VPN			
			1-shot	5-shot	10-shot	20-shot
S+T	21.8M	✗	41.45 ± 0.62	41.85 ± 0.55	47.59 ± 0.48	48.40 ± 0.39
MME	21.8M	✗	40.70 ± 0.71	46.45 ± 0.63	48.45 ± 0.57	57.55 ± 0.44
CDAC	21.8M	✗	35.99 ± 0.83	45.04 ± 0.69	50.45 ± 0.60	54.05 ± 0.51
SHOT++	21.8M	✓	38.90 ± 0.76	39.76 ± 0.68	44.47 ± 0.55	52.60 ± 0.46
Proposed-1	**2.21M**	✓	**46.76 ± 0.82**	**51.08 ± 0.65**	**53.73 ± 0.54**	**65.14 ± 0.37**

Table 3
Accuracy (%) on AndroidVPN.

Method	Params	Source-Free	Normal→NordVPN				Normal→TurboVPN				Normal→SuperVPN			
			1-shot	5-shot	10-shot	20-shot	1-shot	5-shot	10-shot	20-shot	1-shot	5-shot	10-shot	20-shot
S+T	21.8M	✗	0.92 ± 0.18	1.64 ± 0.25	1.56 ± 0.22	2.90 ± 0.31	0.99 ± 0.17	0.80 ± 0.21	1.51 ± 0.26	5.73 ± 0.44	1.28 ± 0.20	1.45 ± 0.24	2.49 ± 0.33	3.77 ± 0.40
MME	21.8M	✗	1.33 ± 0.24	2.57 ± 0.36	2.87 ± 0.41	4.57 ± 0.52	0.60 ± 0.15	2.91 ± 0.38	4.69 ± 0.57	26.23 ± 1.32	1.88 ± 0.27	2.86 ± 0.42	3.78 ± 0.51	9.33 ± 0.75
CDAC	21.8M	✗	1.01 ± 0.20	1.98 ± 0.31	2.38 ± 0.37	3.30 ± 0.45	0.77 ± 0.16	2.36 ± 0.35	16.11 ± 1.04	69.97 ± 1.68	1.71 ± 0.25	2.13 ± 0.34	3.29 ± 0.46	9.63 ± 0.82
SHOT++	21.8M	✓	1.73 ± 0.29	3.50 ± 0.45	4.71 ± 0.53	5.90 ± 0.61	0.97 ± 0.18	5.50 ± 0.62	53.24 ± 1.57	73.33 ± 1.46	1.98 ± 0.30	4.46 ± 0.56	8.24 ± 0.69	17.40 ± 1.05
Proposed-1	**2.21M**	✓	**4.67 ± 0.71**	**22.73 ± 1.36**	**30.87 ± 1.12**	**33.20 ± 0.94**	**3.60 ± 0.58**	**65.13 ± 1.82**	**77.67 ± 1.25**	**87.80 ± 0.91**	**9.06 ± 1.02**	**51.33 ± 1.74**	**53.53 ± 1.46**	**59.13 ± 1.21**

Table 4
Accuracy (%) on real-world environments.

Method	Params	Source-Free	Replay-0→Replay-1				Replay-0→Replay-2				Replay-0→Replay-3			
			1-shot	5-shot	10-shot	20-shot	1-shot	5-shot	10-shot	20-shot	1-shot	5-shot	10-shot	20-shot
S+T	21.8M	✗	73.87 ± 0.51	78.95 ± 0.43	96.84 ± 0.25	97.05 ± 0.18	66.52 ± 0.62	69.83 ± 0.51	87.31 ± 0.34	87.55 ± 0.28	70.06 ± 0.57	93.77 ± 0.32	95.33 ± 0.24	99.14 ± 0.12
MME	21.8M	✗	76.25 ± 0.47	79.79 ± 0.41	92.65 ± 0.31	97.83 ± 0.20	63.45 ± 0.66	66.40 ± 0.55	79.96 ± 0.42	94.92 ± 0.24	76.19 ± 0.49	85.18 ± 0.38	99.38 ± 0.10	99.51 ± 0.08
CDAC	21.8M	✗	64.17 ± 0.71	66.89 ± 0.62	90.59 ± 0.37	92.99 ± 0.31	50.56 ± 0.78	51.24 ± 0.69	79.08 ± 0.45	92.23 ± 0.29	69.01 ± 0.64	69.34 ± 0.58	99.43 ± 0.09	99.57 ± 0.07
SHOT++	21.8M	✓	55.84 ± 0.82	65.40 ± 0.71	91.54 ± 0.35	95.13 ± 0.24	46.04 ± 0.91	52.94 ± 0.72	81.20 ± 0.43	94.32 ± 0.22	73.64 ± 0.55	81.55 ± 0.46	97.65 ± 0.18	99.67 ± 0.06
Proposed-1	**2.21M**	✓	**94.36 ± 0.43**	**95.07 ± 0.36**	**97.04 ± 0.21**	**98.98 ± 0.15**	**81.44 ± 0.67**	**82.88 ± 0.54**	**92.40 ± 0.31**	**98.53 ± 0.20**	**99.59 ± 0.18**	**99.91 ± 0.07**	**99.97 ± 0.04**	**99.98 ± 0.03**

Table 5
Ablation for the offline method.

Filtering Strategy				Momen.	Loss		Average Accuracy (%)		
Adapt. Conf.	Static Conf.	Unc.	DoC	λ_e	$\mathcal{L}_3$	$\mathcal{L}_4$	ISCXVPN 2016	Android VPN	Replay-1/2/3
✗	✗	✗	✗	✗	✗	✗	50.33	29.76	95.28
✗	✗	✗	✗	✗	✗	✓	52.27	30.83	96.95
✓	✗	✓	✓	✓	✓	✗	63.62	56.47	98.51
✗	✓	✓	✓	✓	✓	✓	64.02	57.22	98.73
✓	✗	✓	✓	✗	✓	✗	62.85	55.81	98.42
✓	✗	✓	✓	✓	✓	✓	**65.14**	**60.04**	**99.16**

5.1. Offline adaptation overall performance

5.1.1. Detection accuracy under varying domain shifts

To assess the robustness and adaptability of the offline method, we conducted comprehensive evaluations under varying domain shift scenarios using few-shot settings. On ISCXVPN2016 with a moderate domain shift, Table 2 shows that Proposed-1 consistently outperforms all baselines across all few-shot settings with stable performance. In the 1-shot setting, which constitutes only 0.0245% of the VPN domain data, Proposed-1 achieves a gain of 5.31 percentage points gain over the strongest baseline while using only 10% of its parameter count. This demonstrates the method's strong generalization ability even with scarce labeled samples.

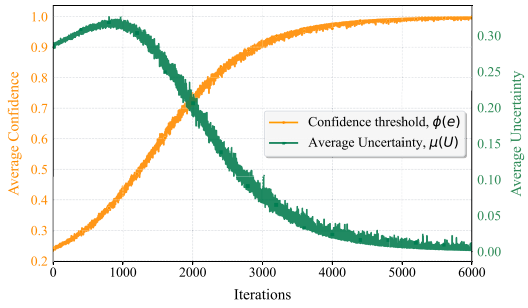
Under conditions of severe domain shift in AndroidVPN, Table 3 shows that standard SSDA methods (e.g., S+T, MME, and CDAC) perform poorly, particularly for Normal-to-NordVPN and Normal-to-SuperVPN tasks. In contrast, SF-SSDA approaches (SHOT++ and Proposed-1) achieve significant improvements. These results highlight the difficulty of conventional joint source–target adaptation under severe domain shift and demonstrate that target-focused source-free adaptation provides an effective alternative in this setting. Meanwhile, Proposed-1 consistently outperforms SHOT++ across different few-shot settings, indicating that reliability-aware pseudo-label optimization provides additional robustness under severe shifts.

Table 4 reports detection performance in real-world replayed environments. While some SSDA methods perform reasonably well under 10-shot and 20-shot settings, Proposed-1 demonstrates stable and accurate threat detection performance even with minimal labeled threat samples (1-shot and 5-shot). This superior performance with scarce threat intelligence is particularly valuable for cybersecurity applications, where newly emerging threats often have limited labeled examples.

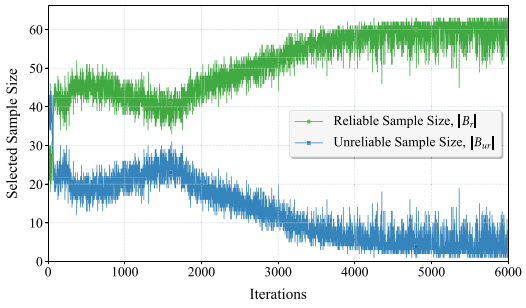
Tables 2–4 reveal that detection accuracy improves steadily as the number of labeled samples increases across all shift scenarios. The relatively small standard deviations further indicate that the performance of Proposed-1 remains stable across different labeled-sample selections and random seeds, rather than being dependent on a particular sampling strategy. Overall, Proposed-1 consistently achieves the highest accuracy, demonstrating its robustness under different domain shift intensities and its ability to mitigate adaptation instability caused by unreliable target-domain supervision.

5.1.2. Convergence analysis

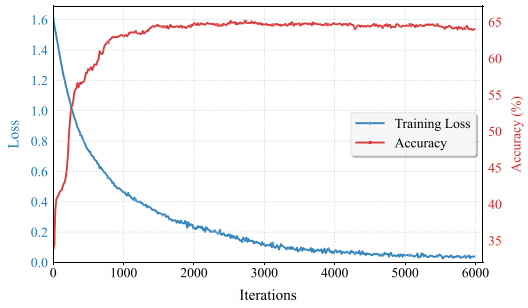
To understand the training dynamics of the offline adaptation framework, we analyze the convergence behavior on the ISCXVPN2016 dataset under the 20-shot setting. As shown in Fig. 5(a), the average prediction confidence increases steadily while the average uncertainty decreases, indicating a progressive improvement in pseudo-label reliability. Fig. 5(b) illustrates a gradual increase in the number of reliable samples during training, with a significant transition observed between iterations 1000 and 2000, where the number of reliable samples increases from around 30 to 50, while unreliable samples decline. This suggests that the filtering mechanism effectively identifies reliable pseudo-labels as training progresses. Fig. 5(c) shows that model accuracy stabilizes at approximately 2000 iterations, while loss only fully stabilizes after about 3000 iterations. This phenomenon



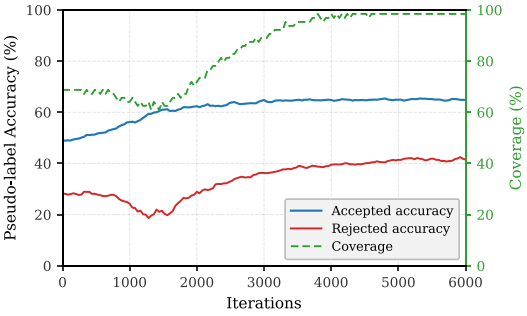
(a) Average confidence and uncertainty scores.



(b) Selected sample size with batch size 64.



(c) Training accuracy and loss.



(d) Pseudo-label accuracy and selection coverage.

Fig. 5. Convergence analysis of offline adaptation on ISCXVPN2016 under the 20-shot setting.

of "early accuracy convergence, delayed loss convergence" suggests that once the model establishes accurate decision boundaries, subsequent optimization focuses on refining the confidence distribution of predictions.

To evaluate the filtering mechanism, Fig. 5(d) reports the pseudo-label accuracy of the accepted and rejected subsets together with the selection coverage. The accepted subset consistently achieves higher pseudo-label accuracy than the rejected subset, while the coverage progressively increases to nearly 100% after an early fluctuation. These results directly demonstrate that the proposed reliability-aware filtering preferentially retains higher-quality pseudo-labels while progressively incorporating more target samples.

5.1.3. Ablation study for offline adaptation

Based on the 20-shot setting of three datasets, we designed ablation experiments to validate the effectiveness of each component in the offline method. As shown in Table 5, using only labeled data leads to significant accuracy degradation, which is particularly severe under AndroidVPN. The iteration-adaptive confidence threshold substantially outperforms the static ones, effectively capturing the reliability of predictions on time-varying network traffic. The reliability-driven momentum coefficient enhances model adaptation performance, whereas removing reconstruction loss ($\mathcal{L}_4$) degrades performance, confirming the essential role of self-supervised signals in regularization. The combination of adaptive confidence and DoC outperforms individual mechanisms, revealing the importance of selecting diverse pseudo-labels. The results confirm that all components are necessary, with performance gains increasing alongside shift intensity.

5.1.4. Effect of backbone and adaptation strategy

To disentangle the effect of adaptation strategy from the backbone architecture and pre-training scheme, we conduct a controlled comparison using Mamba-based variants. All variants use the same Mamba backbone and source-domain MAE-pretrained initialization. Mamba-S directly evaluates the source-trained model on the target domains without target-domain training, whereas Mamba-MT, Mamba-SHOT++, and Proposed-1 use the same 20-shot labeled target splits and unlabeled target data.

Table 6
Average accuracy of matched-backbone Mamba variants on the three AndroidVPN target domains.

Method	Adaptation Strategy	Avg. (%)
Mamba-S	Source-only	3.12
Mamba-MT	Mean Teacher	39.85
Mamba-SHOT++	SHOT++	49.72
Proposed-1	**Reliability-aware**	**60.04**

The average accuracy is reported in Table 6. Mamba-S exhibits limited direct generalization under severe cross-protocol domain shifts. With access to the target-domain data, Mamba-MT and Mamba-SHOT++ improve the average accuracy to 39.85% and 49.72%, respectively, demonstrating the effectiveness of target-domain adaptation under the matched-backbone setting. Proposed-1 achieves the highest average accuracy of 60.04%, outperforming Mamba-MT and Mamba-SHOT++ by 20.19 and 10.32 percentage points, respectively. The improvement over these matched-backbone baselines demonstrates that the performance gain cannot be attributed solely to the Mamba architecture or MAE pre-training and provides direct evidence for the contribution of the proposed reliability-aware adaptation.

5.2. Online adaptation for real-time detection

5.2.1. Detection accuracy for real-world threat traffic

To assess the real-time adaptability of our online method, we performed experiments on real-world replay environments. Specifically, target domain data arrived in mini-batch streams, with each sample observed only once. Initially, model parameter updates and pseudo-label refinement were disabled to simulate cold-start conditions; the refinement mechanism was then activated once the buffer queue accumulated 128 statistical entries.

As shown in Table 7, we report the mean accuracy and standard deviation over five independent runs. Proposed-2 achieved $72.08 \pm 0.31\%$, $67.55 \pm 0.41\%$, and $90.95 \pm 0.29\%$ on Replay-1, -2, and -3, respectively, outperforming AdaContrast by 0.77, 5.60, and 14.77 percentage points.

Table 7
Online adaptation accuracy (%, mean ± std. over five runs) on real-world environments.

Method	Params	Replay-0→-1	Replay-0→-2	Replay-0→-3
AdaContrast	21.8M	71.31±0.46	61.95±0.58	76.18±0.52
Proposed-3	2.83M	71.74±0.39	66.72±0.47	89.84±0.36
Proposed-2	**2.21M**	**72.08±0.31**	**67.55±0.41**	**90.98±0.29**

Table 8
Online detection accuracy (%, mean ± std. over five runs) under sequential target-domain shifts.

Method	Backbone	Replay-0→-1	Replay-1→-2	Replay-2→-3
AdaContrast	ResNet	71.29±0.51	57.84±1.42	68.31±1.76
Proposed-3	Transformer	71.81±0.46	60.76±1.28	75.92±1.55
Proposed-2	Mamba	**72.12±0.43**	**62.41±1.16**	**78.63±1.39**

Although the gain on Replay-1 is relatively small, the repeated-run results indicate stable performance across random seeds. Proposed-3 (Transformer) also outperformed AdaContrast but showed slightly lower accuracy than Proposed-2. Considering latency and model size, Proposed-2 offers a better trade-off for real-time detection.

Note that online evaluation is conducted exclusively on real-world binary threat-detection tasks due to the constraint of a fixed label space. In contrast, offline experiments use multi-class application traffic datasets.

5.2.2. Sequential online adaptation across changing target environments

To further evaluate online adaptation under changing deployment conditions, we construct a non-stationary traffic stream by sequentially concatenating Replay-1, Replay-2, and Replay-3. The model is initialized from Replay-0 only once, with the student and teacher parameters, optimizer state, and FIFO statistics queue continuously carried across transitions. No target labels, environment identities, or transition boundaries are available during adaptation.

We adopt a strict prediction-before-update protocol. For each incoming mini-batch, predictions are first generated by the current model and recorded for evaluation. The same unlabeled mini-batch is then used for one adaptation step before processing the subsequent mini-batch. Ground-truth labels are used exclusively for performance evaluation and are never involved in model updating. All experiments are repeated five times with different random seeds, and the mean accuracy and standard deviation are reported.

As shown in Table 8, the performance varies across successive target environments. Replay-0→-1 remains close to the corresponding independent-domain result, while a clear performance drop is observed after the transition to Replay-2. The accuracy subsequently recovers on Replay-3, although it remains lower than that obtained when Replay-3 is adapted independently from Replay-0. These results indicate that adaptation states accumulated from preceding domains can influence subsequent online adaptation. Nevertheless, Proposed-2 consistently achieves the highest accuracy across all target environments, demonstrating stronger robustness to sequential target-distribution shifts.

5.2.3. Ablation study for online adaptation

Table 9 reports ablation results under online streaming conditions. Without adaptation, the model achieves only 61.78% accuracy, highlighting the challenge of single-pass learning without batch-level statistics. The buffer queue with adaptive thresholding addresses this limitation by approximating batch statistics from a small queue, achieving 73.12% accuracy while maintaining bounded memory consumption. Removing DoC causes accuracy to drop to 72.23%, as the model becomes biased toward dominant categories under limited queue observations. The reconstruction loss ($\mathcal{L}_4$) and adaptive momentum contribute complementary gains, jointly boosting accuracy to 76.86%. These results demonstrate that the proposed components transfer from batch to streaming settings while maintaining computational efficiency.

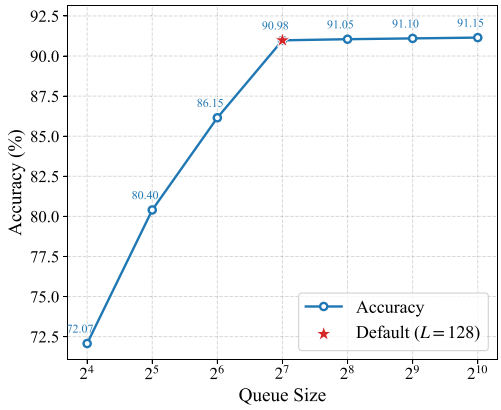


Fig. 6. Accuracy vs. queue size.

5.2.4. Sensitivity to buffer queue length

We further analyze the impact of buffer queue length on online adaptation performance. The sensitivity analysis was conducted on Replay-0 to Replay-3, with the queue length set to $\{16, 32, \ldots, 1024\}$. As shown in Fig. 6, the average accuracy generally improves as the queue length increases, reaching its highest value of 91.15% at $L = 1024$.

With $L = 128$, which corresponds to only 0.43% of the complete 30,000-flow stream, the online method already achieves an average accuracy of 90.98%, only 0.17 percentage points below the highest accuracy obtained with $L = 1024$. Therefore, we set $L = 128$ by default, as it provides sufficient recent statistics with a short initialization period and near-optimal adaptation performance. The FIFO queue stores four FP32 scalar statistics for each entry: mean confidence, uncertainty, cross-view dispersion, and DoC. Its theoretical raw storage cost is therefore $O(L)$ and is independent of the total stream length. Specifically, these four statistics require $16L$ bytes, corresponding to approximately 2 KiB for $L = 128$ and 16 KiB for $L = 1024$.

5.3. Inference efficiency

5.3.1. Offline method

Given the scale and frequency of encrypted traffic, inference latency is as critical as accuracy for real-world deployment. Using the 20-shot setting on ISCXVPN2016 with a batch size of 64, we benchmark inference efficiency across all methods. As shown in Fig. 7(a), methods based on ResNet-34 (S+T, MME, CDAC, SHOT++) suffer from slower inference due to heavy convolution operations. In contrast, Proposed-1, benefiting from efficient linear sequence modeling, achieves an average inference time of just 2.175 s. Fig. 7(b) shows that SSDA methods incur high memory overhead due to the need for source–target feature alignment and source data access. In comparison, source-free methods (SHOT++ and Proposed-1) eliminate the need to store source data, significantly improving memory efficiency.

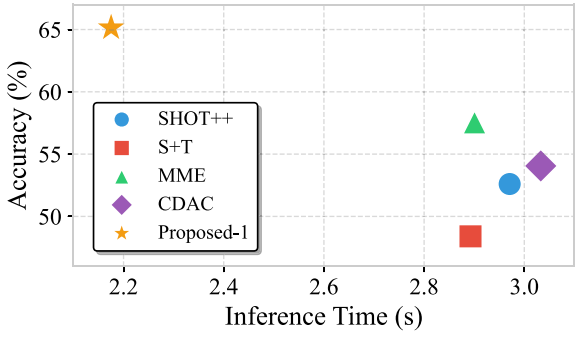
5.3.2. Online method

For the OTTA setting, which involves both continuous adaptation and online inference, we measure inference and adaptation time over the complete streaming sequence of 30,000 flows. The inference time denotes the forward-pass cost for generating predictions, while the adaptation time represents the cost of model parameter updates over the incoming mini-batches.

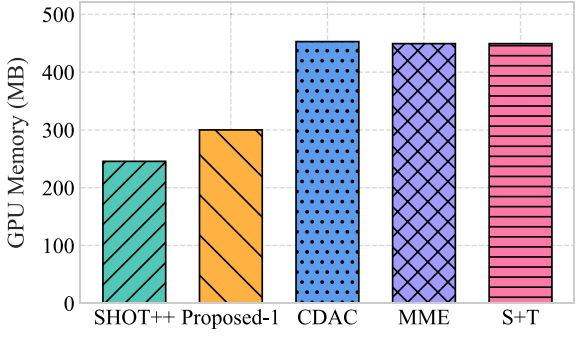
As shown in Fig. 8(a), Proposed-2 requires only 0.64 s for inference and 7.4 s for adaptation, achieving a 2× speedup over AdaContrast in total online processing time. Although the Transformer version (Proposed-3) is limited by the quadratic complexity of self-attention, it still achieves lower inference latency than AdaContrast. To further evaluate the deployment efficiency, the accumulated latency is normalized by the 30,000 processed flows, and the results are reported in Table 10.

Table 9
Ablation for online method on real-world environments.

Filtering Strategy					Momen.	Loss	Avg. (%)
Small-scale Queue	Adapt. Conf.	Static Conf.	Unc.	DoC	λ_q	$\mathcal{L}_4$	
✗	✗	✗	✗	✗	✗	✗	61.78
✓	✗	✓	✓	✓	✓	✓	72.45
✓	✓	✗	✓	✓	✓	✗	73.12
✓	✓	✗	✓	✓	✗	✓	74.35
✓	✓	✗	✓	✗	✓	✗	72.23
✓	✓	✗	✓	✓	✓	✓	**76.86**

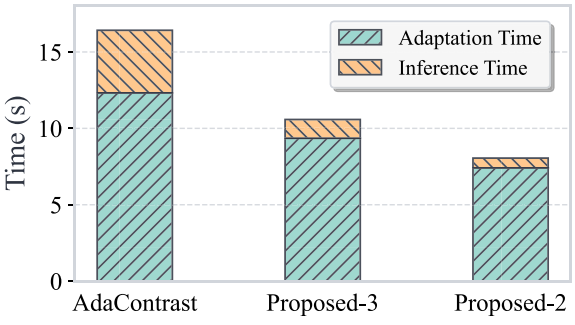


(a) Inference vs. accuracy.

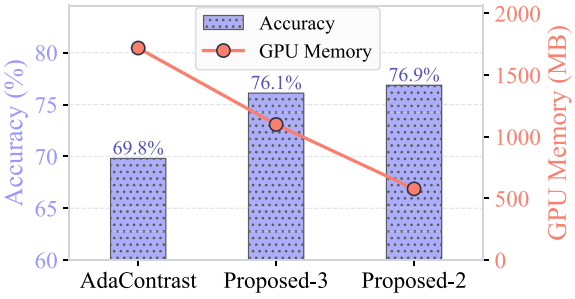


(b) GPU memory.

Fig. 7. Offline adaptation efficiency.



(a) Inference vs. adaptation.



(b) Average accuracy vs. GPU memory.

Fig. 8. Online adaptation efficiency.

Table 10
Normalized online latency and throughput comparison.

Method	Inference (ms/flow)	Adaptation (ms/flow)	Throughput (flows/s)
AdaContrast	0.1367	0.4117	1827
Proposed-3	0.0410	0.3128	2836
Proposed-2	**0.0215**	**0.2480**	**3725**

Beyond computational efficiency, we further evaluate the memory consumption of different online adaptation methods. As shown in Fig. 8(b), the GPU memory usage of Proposed-3 and Proposed-2 is 1087 MB and 578 MB, respectively, reducing memory consumption by 37.5% and 67.2% compared with AdaContrast. Meanwhile, Proposed-2 and Proposed-3 achieve average accuracies of 76.86% and 76.10%, respectively, demonstrating that the proposed framework maintains competitive adaptation performance while improving computational efficiency.

6. Conclusion

This paper develops a Mamba-based self-training framework for cross-domain encrypted traffic detection under heterogeneous distribution shifts. Following a source-free adaptation paradigm, the framework supports both offline semi-supervised adaptation and online unsupervised adaptation through reliability-aware pseudo-label filtering. The offline method achieves consistently strong detection performance across different shift intensities while using only about 10% of the parameters required by the ResNet-based baselines. For streaming threat traffic, the batch-based filtering mechanism is extended with a fixed-size FIFO statistics queue, enabling lightweight online adaptation with bounded memory overhead. Experiments under both independent and sequentially changing environments further demonstrate the effectiveness of the online method. An architectural comparison with a Transformer-based variant shows that Mamba offers a better balance of detection accuracy, inference efficiency, and model complexity. Beyond encrypted traffic detection, the framework design and source-free paradigm enable straightforward customization for IoT anomaly detection and adaptive malware classification.

CRediT authorship contribution statement

Qi Liu: Writing – original draft, Visualization, Methodology, Investigation. **Hang Shen:** Writing – review & editing, Writing – original draft, Supervision, Resources, Methodology, Investigation, Funding acquisition, Conceptualization. **Tianjing Wang:** Supervision, Methodology, Funding acquisition. **Guangwei Bai:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Xinchen Zhang, Running Zhao, Zhihan Jiang, Handi Chen, Yulong Ding, Edith C.H. Ngai, Shuang-Hua Yang, Continual learning with strategic selection and forgetting for network intrusion detection, in: Proc. IEEE Int. Conf. Comput. Commun., INFOCOM, 2025, pp. 1–10.

[2] Hassan Iqbal, Anand Singh, Muhammad Shahzad, Characterizing the availability and latency in AWS network from the perspective of tenants, IEEE/ACM Trans. Netw. 30 (4) (2022) 1554–1568.

[3] Renjie Xie, Jiahao Cao, Enhuan Dong, Mingwei Xu, Kun Sun, Qi Li, Licheng Shen, Menghao Zhang, Rosetta: Enabling robust TLS encrypted traffic classification in diverse network environments with TCP-aware traffic augmentation, in: Proc. USENIX Security Symp., 2023, pp. 625–642.

[4] Gerard Draper-Gil, Arash Habibi Lashkari, Mohammad Saiful Islam Mamun, Ali A. Ghorbani, Characterization of encrypted and VPN traffic using time-related features, in: Proc. Int. Conf. Inf. Syst. Secur. Privacy, ICISSP, 2016, pp. 407–414.

[5] Sanghak Oh, Minwook Lee, Hyunwoo Lee, Elisa Bertino, Hyoungshick Kim, AppSniffer: Towards robust mobile App fingerprinting against VPN, in: Proc. ACM Web Conf., WWW, 2023, pp. 2318–2328.

[6] Kwizera K. Jonath, Abida Naz, Shigeng Zhang, GMM-cGAN: mitigating data scarcity and label noise for robust encrypted malicious traffic classification, Comput. Netw. 274 (2026) 111823.

[7] Vincent F. Taylor, Riccardo Spolaor, Mauro Conti, Ivan Martinovic, Robust smartphone App identification via encrypted network traffic analysis, IEEE Trans. Inf. Forensics Secur. 13 (1) (2018) 63–78.

[8] Susu Cui, Cong Dong, Meng Shen, Yuling Liu, Bo Jiang, Zhigang Lu, CBSeq: A channel-level behavior sequence for encrypted malware traffic detection, IEEE Trans. Inf. Forensics Secur. 18 (2023) 5011–5025.

[9] Yuming Feng, Yu Zhang, Hui He, Weizhe Zhang, Desheng Wang, An IoT device identification method using extracted fingerprint from sequence of traffic grayscale images, IEEE Trans. Dependable Secur. Comput. 21 (6) (2024) 5737–5754.

[10] Ting-Li Huoh, Yan Luo, Peilong Li, Tong Zhang, Flow-based encrypted network traffic classification with graph neural networks, IEEE Trans. Netw. Serv. Manag. 20 (2) (2023) 1224–1237.

[11] Zijie Chen, Long Wen, Hailin Zou, Chenyang Zhang, Binbin Wang, Jianqing Li, Yuanyuan Pan, EMLF-ETD: An efficient multi-level feature representation approach for encrypted traffic detection in variable-length traffic sessions, Comput. Netw. 286 (2026) 112499.

[12] Matan Klein, Revital Marbel, Chen Hajaj, Ran Dubin, Amit Dvir, GraphMux: A graph-based framework for encrypted traffic classification, Comput. Netw. 285 (2026) 112370.

[13] Xinjie Lin, Gang Xiong, Gaopeng Gou, Wenqi Dong, Jing Yu, Zhen Li, Wei Xia, Respond to change with constancy: Instruction-tuning with LLM for Non-I.I.D. network traffic classification, IEEE Trans. Inf. Forensics Secur. 20 (2025) 5758–5773.

[14] Huu-Phong Phan-Nguyen, Anh Dao, Tien-Huy Nguyen, Tuan Quang, Huu-Loc Tran, Tinh-Anh Nguyen-Nhu, Huy-Thach Pham, Quan Nguyen, Hoang Le, Quang-Vinh Dinh, Cycle training with semi-supervised domain adaptation: Bridging accuracy and efficiency for real-time mobile scene detection, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., CVPR, 2025, pp. 1888–1897.

[15] Xuejiao Luo, Pengcheng Wei, Wenyin Liu, Jiahao Cao, Jiang Li, Binbin Chen, GM-BFD: A generalizable multi-view framework for botnet flow detection, in: Proc. IEEE Int. Conf. Comput. Commun., INFOCOM, 2025, pp. 1–10.

[16] Jingjing Li, Zhiqi Yu, Zhekai Du, Lei Zhu, Heng Tao Shen, A comprehensive survey on source-free domain adaptation, IEEE Trans. Pattern Anal. Mach. Intell. 46 (8) (2024) 5743–5762.

[17] Gezheng Xu, Li Yi, Pengcheng Xu, Jiaqi Li, Ruizhi Pu, Changjian Shui, A. Ian McLeod, Boyu Wang, Charles Ling, Unraveling the mysteries of label noise in source-free domain adaptation: Theory and practice, IEEE Trans. Pattern Anal. Mach. Intell. 47 (5) (2025) 4007–4023.

[18] Fuming You, Jingjing Li, Lei Zhu, Zhi Chen, Zi Huang, Domain adaptive semantic segmentation without source data, in: Proc. ACM Int. Conf. Multimedia, MM, 2021, pp. 3293–3302.

[19] Jiayi Tian, Jing Zhang, Wen Li, Dong Xu, VDM-DA: Virtual domain modeling for source data-free domain adaptation, IEEE Trans. Circuits Syst. Video Technol. 32 (6) (2022) 3749–3760.

[20] Shiqi Yang, Yaxing Wang, Joost Van De Weijer, Luis Herranz, Shangling Jui, Generalized source-free domain adaptation, in: Proc. IEEE/CVF Int. Conf. Comput. Vis., ICCV, 2021, pp. 8978–8987.

[21] Haojin Li, Heng Li, Jianyu Chen, Rihan Zhong, Ke Niu, Huazhu Fu, Jiang Liu, AIF-SFDA: Autonomous information filter driven source-free domain adaptation for medical image segmentation, in: Proc. AAAI Conf. Artif. Intell., 2025, pp. 4716–4724.

[22] Kate Saenko, Brian Kulis, Mario Fritz, Trevor Darrell, Adapting visual category models to new domains, in: Proc. Eur. Conf. Comput. Vis., ECCV, 2010, pp. 213–226.

[23] Zhiqi Yu, Zhichao Liao, Jingjing Li, Zhi Chen, Lei Zhu, Dynamic target distribution estimation for source-free open-set domain adaptation, in: Proc. AAAI Conf. Artif. Intell., AAAI, 2025.

[24] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, Kate Saenko, Semi-supervised domain adaptation via minimax entropy, in: Proc. IEEE/CVF Int. Conf. Comput. Vis., ICCV, 2019, pp. 8050–8058.

[25] Jichang Li, Guanbin Li, Yemin Shi, Yizhou Yu, Cross-domain adaptive clustering for semi-supervised domain adaptation, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., CVPR, 2021, pp. 2505–2514.

[26] Chao Ren, Jinbo Chen, Rui Li, Weihao Zheng, Yijiang Chen, Yikun Yang, Xiaowei Zhang, Bin Hu, Semi-supervised pairwise transfer learning based on multi-source domain adaptation: A case study on EEG-based emotion recognition, Knowl.-Based Syst. 305 (2024) 112669.

[27] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, Jiashi Feng, Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer, IEEE Trans. Pattern Anal. Mach. Intell. 44 (11) (2022) 8602–8617.

[28] Ning Ma, Haishuai Wang, Zhen Zhang, Sheng Zhou, Hongyang Chen, Jiajun Bu, Source-free semi-supervised domain adaptation via progressive mixup, Knowl.-Based Syst. 262 (2023) 110208.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., CVPR, 2016, pp. 770–778.

[30] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proc. Int. Conf. Learn. Represent., ICLR, 2015, pp. 1–14.

[31] Shengcao Cao, Dhiraj Joshi, Liang-Yan Gui, Yu-Xiong Wang, Contrastive mean teacher for domain adaptive object detectors, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., CVPR, 2023, pp. 23839–23848.

[32] Xingguang Zhang, Chih-Hsien Chou, Source-free domain adaptation for video object detection under adverse image conditions, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., CVPR, 2024, pp. 5010–5019.

[33] Ankit Singh, CLDA: Contrastive learning for semi-supervised domain adaptation, in: Proc. Adv. Neural Inf. Process. Syst., NIPS, 2021, pp. 5089–5101.

[34] Jian Liang, Dapeng Hu, Jiashi Feng, Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation, in: Proc. Int. Conf. Mach. Learn., ICML, 2020, pp. 6028–6039.

[35] Nazmul Karim, Niluthpol Chowdhury Mithun, Abhinav Rajvanshi, Han-pang Chiu, Supun Samarasekera, Nazanin Rahnavard, C-SFDA: A curriculum learning aided self-training framework for efficient source-free domain adaptation, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., CVPR, 2023, pp. 24120–24131.

[36] Zixin Wang, Yadan Luo, Zhi Chen, Sen Wang, Zi Huang, Cal-SFDA: Source-free domain-adaptive semantic segmentation with differentiable expected calibration error, in: Proc. ACM Int. Conf. Multimedia, MM, 2023, pp. 1167–1178.

[37] Fuming You, Jingjing Li, Zhi Chen, Lei Zhu, Pixel exclusion: Uncertainty-aware boundary discovery for active cross-domain semantic segmentation, in: Proc. ACM Int. Conf. Multimedia, MM, 2022, pp. 1866–1874.

[38] Chunming He, Rihan Zhang, Fengyang Xiao, Chengyu Fang, Longxiang Tang, Yulun Zhang, Linghe Kong, Deng-Ping Fan, Kai Li, Sina Farsiu, RUN: Reversible unfolding network for concealed object segmentation, 2025, arXiv preprint arXiv:2501.18783.

[39] Chunming He, Kai Li, Yachao Zhang, Ziyun Yang, Youwei Pang, Longxiang Tang, Chengyu Fang, Yulun Zhang, Linghe Kong, Xiu Li, Sina Farsiu, Segment concealed objects with incomplete supervision, IEEE Trans. Pattern Anal. Mach. Intell. 47 (9) (2025) 7832–7851.

[40] Chunming He, Kai Li, Guoxia Xu, Jiangpeng Yan, Longxiang Tang, Yulun Zhang, Yaowei Wang, Xiu Li, HQG-Net: Unpaired medical image enhancement with high-quality guidance, IEEE Trans. Neural Netw. Learn. Syst. 35 (12) (2023) 18404–18418.

[41] Van Tong, Cuong Dao, Hai-Anh Tran, Duc Tran, Huynh Thi Thanh Binh, Thang Hoang-Nam, Truong X. Tran, Encrypted traffic classification through deep domain adaptation network with smooth characteristic function, IEEE Trans. Netw. Serv. Manag. 22 (1) (2025) 331–343.
[42] Zixin Wang, Yadan Luo, Liang Zheng, Zhuoxiao Chen, Sen Wang, Zi Huang, In search of lost online test-time adaptation: A survey, Int. J. Comput. Vis. (IJCV) 133 (3) (2025) 1106–1139.
[43] David Osowiechi, Gustavo A. Vargas Hakim, Mehrdad Noori, Milad Cheraghalikhani, Ali Bahri, Moslem Yazdanpanah, Ismail Ben Ayed, Christian Desrosiers, NC-TTT: A noise contrastive approach for test-time training, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., CVPR, 2024, pp. 6078–6086.
[44] Yossi Gandelsman, Yu Sun, Xinlei Chen, Alexei Efros, Test-time training with masked autoencoders, in: Proc. Adv. Neural Inf. Process. Syst., Vol. 35, NIPS, 2022, pp. 29374–29385.
[45] Albert Gu, Tri Dao, Mamba: Linear-time sequence modeling with selective state spaces, in: Proc. 1st Conf. Lang. Model., 2024, URL https://openreview.net/forum?id=tEYskw1VY2.
[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, Attention is all you need, in: Proc. Adv. Neural Inf. Process. Syst., NIPS, 2017, pp. 6000–6010.
[47] Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, Ludwig Schmidt, Predicting with confidence on unseen distributions, in: Proc. IEEE/CVF Int. Conf. Comput. Vis., ICCV, 2021, pp. 1134–1144.
[48] Subhankar Roy, Martin Trapp, Andrea Pilzer, Juho Kannala, Nicu Sebe, Elisa Ricci, Arno Solin, Uncertainty-guided source-free domain adaptation, in: Proc. Eur. Conf. Comput. Vis., ECCV, 2022, pp. 537–555.
[49] Jason A. Donenfeld, WireGuard: Next generation kernel network tunnel, in: Proc. ISOC Netw. Distrib. Syst. Secur. Symp., NDSS, 2017, pp. 1–12.
[50] Ruijie Zhao, Mingwei Zhan, Xianwen Deng, Fangqi Li, Yanhao Wang, Yijun Wang, Guan Gui, Zhi Xue, A novel self-supervised framework based on masked autoencoder for traffic classification, IEEE/ACM Trans. Netw. 32 (3) (2024) 2012–2025.
[51] Tongze Wang, Xiaohui Xie, Wenduo Wang, Chuyi Wang, Youjian Zhao, Yong Cui, Netmamba: Efficient network traffic classification via pre-training unidirectional mamba, in: Proc. IEEE Int. Conf. Network Protocols, ICNP, 2024, pp. 1–11.
[52] Dian Chen, Dequan Wang, Trevor Darrell, Sayna Ebrahimi, Contrastive test-time adaptation, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., CVPR, 2022, pp. 295–305.



Qi Liu received the B.Eng. degree in Computer Science from Shenyang Ligong University, Shenyang, China, in 2023, and the M.Eng. degree in Computer Science from Nanjing Tech University, Nanjing, China, in 2026. He is currently pursuing the Ph.D. degree in Cyberspace Security at Southeast University, Nanjing, China. His research interests include unsupervised, semi-supervised, and continual learning in cybersecurity, as well as mobile vision computing in the Internet of Drones and Vehicles. He is an IEEE Graduate Student Member and a CCF Student Member. He has published papers in Elsevier Comput. Netw., IEEE TITS, IEEE IoT-J, and IEEE TCE.



Hang Shen received the Ph.D. degree (with honors) in Computer Science from Nanjing University of Science and Technology, Nanjing, China, in 2015. He was a Postdoctoral Fellow with the Broadband Communications Research (BBCR) Lab, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, from 2018–2019. He is an Associate Professor at Nanjing Tech University, Nanjing, China. His research interests include vehicular networks and cybersecurity with agentic AI. He serves as an Associate Editor for J. Inf. Process., Front. Blockchain, and IEEE Access, and has served as a Guest Editor for Peer-to-Peer Netw. Appl. He serves/served as a TPC member of IEEE GLOBECOM, IEEE SVCC, IEEE ISPA, IEEE HPCC, IEEE CCECE, PST, and ICA3PP. He is a member of IEEE, an Executive Committee Member of the ACM Nanjing Chapter, and a CCF Senior Member.



Tianjing Wang holds a B.Sc. in Mathematics at the Nanjing Normal University in 2000, an M.Sc. in Mathematics at Nanjing University in 2002, and a Ph.D. in Signal and Information System at the Nanjing University of Posts and Telecommunications in 2009. From 2011 to 2013, she was a Full-Time Postdoctoral Fellow with the School of Electronic Science and Engineering, Nanjing University of Posts and Telecommunications. From 2013 to 2014, she was a Visiting Scholar with the Electrical and Computer Engineering Department at the State University of New York at Stony Brook. She is an Associate Professor with the Communication Engineering Department at Nanjing Tech University, Nanjing, China. Her research interests include agentic AI for cybersecurity and V2X communication networks. He has published papers in Elsevier Comput. Netw., Elsevier JSA, IEEE TMC, IEEE TITS, IEEE TCSS, IEEE TBC, IEEE IoT-J, and IEEE TCE. She is a member of IEEE.



Guangwei Bai received the B.Eng. and M.Eng. degrees in computer engineering from Xi'an Jiaotong University, Xi'an, China, in 1983 and 1986, respectively, and the Ph.D. degree in Computer Science from the University of Hamburg, Hamburg, Germany, in 1999. From 1999 to 2001, he worked as a Research Scientist at the German National Research Center for Information Technology, Bonn, Germany. In 2001, he joined the University of Calgary, Calgary, AB, Canada, as a Research Associate. Since 2005, he has been working at Nanjing Tech University, Nanjing, China, as a Professor in Computer Science. From October to December 2010, he was a Visiting Professor with the Electrical and Computer Engineering Department at the University of Waterloo, Waterloo, ON, Canada. His research interests include future networks, cybersecurity, and agentic AI. He has authored and co-authored more than 90 peer review papers in international journals and conferences. He is a CCF Distinguished Member.