INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

VOL 04. ISSUE 02 . (2026)

http://ijirjournal.com

Multi-Scale Temporal Graph Transformer for Blockchain Anti-Money Laundering

Isaac Edwards [1], Benjamin Murphy *[1], Zain Sharma 1, Diego Lee [2]

[1] Department of Computer Science, University of Texas at Austin, USA

[2] Department of Computer Science, North Carolina State University, USA

Corresponding author: *Benjamin Murphy*

Email: *ben.m8492@utexas.edu*

Abstract.

Money laundering on public blockchains is a temporal pattern: automated peeling moves value within minutes, coordinated deposits arrive within the same quarter of an hour, and layering drips funds through intermediaries over days, while licit activity produces the same shapes at other speeds. We present MS-TGT, a multi-scale temporal graph transformer for subgraph-level anti-money laundering that builds on the time-biased subgraph transformer STEMP. MS-TGT describes every account by a multi-scale flow spectrum, i.e., Laplace-domain samples of its residence-time, synchrony and dormancy distributions at six scales from five minutes to four weeks; it biases attention with a signed mixture of exponential time kernels that can express band-pass and causal preferences; and it reads out the subgraph through scale-specific tokens fused by a gate. An analysis of 79,161 real Ethereum phishing transfers shows residence times spread from seconds to months. On a benchmark calibrated with these timings, in which every laundering motif has licit look-alikes with identical topology and amounts, MS-TGT reaches an illicit-class F1 of 0.949 against 0.815 for the strongest of ten baselines, with bootstrap and McNemar tests confirming the difference, and it degrades least under slowed or delayed laundering. Ablations attribute most of the gain to the flow spectrum, which also lifts GIN from 0.690 to 0.875. On a real-data probe with randomized reference models the gains are smaller and explicit gap statistics remain competitive.

Keywords: *anti-money laundering, blockchain, graph transformer, temporal graph learning, multi-scale modeling, subgraph classification*

1. INTRODUCTION

Public blockchains record every transfer in an append-only ledger [1]. This transparency has turned anti-money laundering (AML) on cryptocurrencies into a graph-learning problem: analysts follow value through chains of addresses and must decide whether a candidate flow is consistent with laundering. Early benchmarks framed the task as node classification on transaction graphs [2], [3]. The Elliptic2 release [4] moved the unit of prediction to the subgraph, in line with the observation that laundering is expressed by the shape of a flow rather than by an individual transaction, and measurement studies of mixing and laundering services report recurring shapes such as peeling chains and scatter–gather structures [5], [6]. Shape alone is ambiguous, however. A peeling chain produced by an automated laundering script and the change outputs of an ordinary wallet can be topologically identical; what separates them is when the hops occur. The relevant timing information also lives at different scales in

different patterns. Automated peeling moves funds within minutes, coordinated smurf deposits arrive within the same quarter of an hour, layering schemes drip value through intermediaries on a daily schedule, and mule accounts may stay dormant for weeks before a burst of activity. Real flows confirm this picture. On 1,259 Ethereum phishing accounts [7], the time between an incoming deposit and the next outgoing transfer is shorter than five minutes in 10.3% of the cases, between one hour and one day in 31.3%, and longer than one week in 17.0% (Fig. 2). A detector therefore has to read temporal structure at several resolutions at once. Existing models cover only part of this requirement. Message-passing graph neural networks (GNNs) [8]–[10] and their directed-multigraph extensions [11] capture flow topology but treat time, if at all, as one more feature. Continuous-time models such as TGAT [12] and TGN [13] encode timestamps but output node-level embeddings. Graph transformers [14], [15] relieve the over-squashing of message passing [16] through global attention, yet they are built for static graphs. The subgraph-level temporal graph transformer STEMP of Lin *et al.* [17] is the closest prior work. It biases attention logits jointly by shortest-path distance and by a learnable function of the time gap between nodes, reads out a subgraph embedding through a virtual token, and explains decisions with laundering-motif prototypes. Its temporal bias, however, is a single monotone function of the absolute gap between node representative times. Such a bias cannot prefer partners that are active about one day apart while ignoring both near-simultaneous and month-apart partners, it cannot tell whether a partner acted before or after a node, and one representative time per node discards how quickly value leaves an account after it arrives. This paper builds on the STEMP design and extends its temporal reasoning to multiple scales. We propose MS-TGT, a multi-scale temporal graph transformer for subgraph-level blockchain AML, and make four contributions. (1) A *multi-scale flow spectrum* summarizes, for every account, the distributions of residence time, arrival and departure synchrony, and dormancy by their Laplace transforms sampled on a geometric grid of time scales from five minutes to four weeks. It can be read as a soft temporal-motif census [18] evaluated at six windows at once, and it enters the model as part of the node token. (2) A multi-scale signed kernel bias replaces STEMP's monotone time-gap bias: each attention head mixes exponential kernels of different widths with separate weights for earlier and later partners, which yields band-pass and causal temporal preferences. (3) *Scale-specific readout tokens* fused by a learned gate expose the time band on which each decision rests. (4) We evaluate MS-TGT on TempoLaunder, a benchmark whose background timing is resampled from real Ethereum transfers and whose laundering motifs have licit look-alikes with identical topology and amount distributions, and on a real-data probe that asks whether a model can distinguish real Ethereum phishing flows from randomized reference models [19] that destroy temporal structure at a chosen scale. Following STEMP's evaluation protocol, we report exact paired tests, instance-level bootstrap intervals and McNemar tests. On TempoLaunder, MS-TGT raises the illicit-class F_1 from 0.815 for the strongest of ten baselines to 0.949, and it keeps 0.886 when every laundering hop is delayed by up to four hours, where the best baseline falls to 0.678. Ablations identify the flow spectrum as the main source of the gain, while, in a single-seed ablation, the signed multi-scale bias and the scale-specific readout leave i.i.d. accuracy unchanged but matter under temporal shift. On real Ethereum data the picture is more nuanced: MS-TGT is the strongest neural model on the probe tasks, but gradient boosting on explicit gap statistics remains better at detecting the loss of sub-hour structure. The remainder of the paper reviews related work (Section II), analyses real timing data (Section III), presents MS-TGT (Section IV) and the experimental protocol (Section V), and reports and discusses the results (Sections VI and VII).

II. Related Work

A. Graph Learning for Cryptocurrency AML

Weber *et al.* [2] released the Elliptic transaction graph and established GCN baselines for illicit-transaction classification, and Elliptic++ [3] added wallet-address graphs. Bellei *et al.* [4] argued that AML is inherently a subgraph problem and released 122K labelled subgraphs of Bitcoin clusters. Egressy *et al.* [11] showed that ego identifiers, port numbering and reverse message passing make message-passing GNNs provably able to detect directed multigraph patterns, with large gains on the synthetic AML data of [20]. FlowScope [21] detects dense multi-step flows without supervision, and Wu *et al.* [7] combined amount- and time-aware random walks with network embedding to detect Ethereum phishing accounts. STEMP unified several of these ideas in a subgraph-level temporal transformer with motif prototypes and cross-chain transfer, and its evaluation protocol, which replaces asymptotic significance claims by exact paired tests and instance-level tests, is the protocol we adopt. Temporal Transformer learning with conditional tabular generation offers a relevant experimental design for capturing transaction sequences while addressing highly imbalanced fraud data [31].

B. Temporal Graph Learning and Temporal Networks

Discrete-time models evolve GNN parameters across snapshots [22]. Continuous-time models encode timestamps with functional encodings motivated by Bochner's theorem [12] or Time2Vec [23] and maintain node memories [13]. Temporal network science contributes complementary tools: time-respecting paths and inter-event statistics [24], bursty heavy-tailed activity [25], temporal motifs counted within a time window δ [18], and randomized reference models that isolate the effect of temporal correlations by shuffling event times under constraints [19]. Temporal-motif counting needs a hand-chosen window; our flow spectrum evaluates soft versions of such counts at several windows simultaneously and lets the network weight them.

C. Graph Transformers and Multi-Scale Modeling

Graphormer [14] injects structure into self-attention [26] through biases indexed by shortest-path distance, and GraphGPS [15] combines local message passing, global attention and positional encodings. STEMP adds a time-gap term to the shortest-path bias. Multi-resolution analysis decomposes a signal into components at geometrically spaced scales [27]. We transfer this idea from the signal to the attention bias: a bank of exponential kernels at geometric scales is shared by all heads, and differences of neighbouring kernels act as band-pass filters over time gaps.

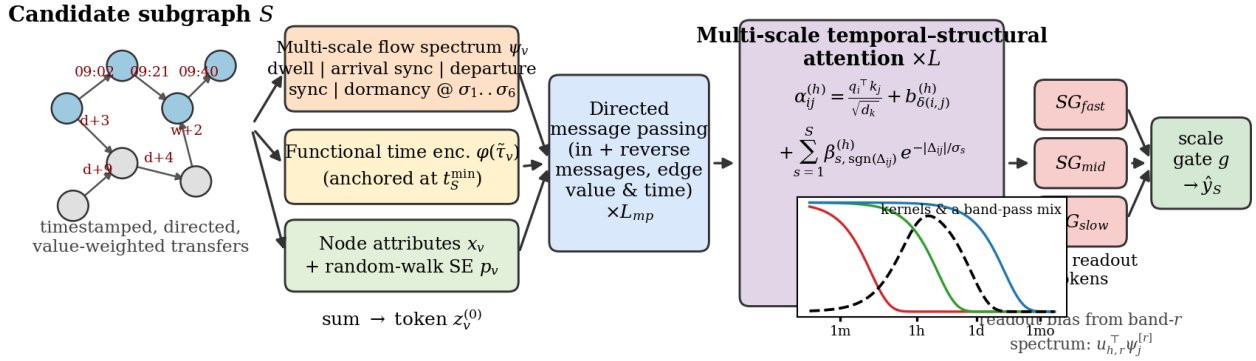


Figure 1Overview of MS-TGT. Node tokens combine attributes, a functional time encoding, a random-walk structural encoding and the multi-scale flow spectrum ψv; directed message passing adds local flow structure; attention logits are biased by shortest-path dista

III. Problem Setting and Empirical Motivation

A. Temporal Subgraph Classification

A candidate subgraph $S=(V_S,E_S)$ consists of accounts (or address clusters) and directed transfers $e=(u,v,t,a)$ that move value $a>0$ from u to v at time t; several transfers may connect the same pair. As in STEMP, times are measured relative to the first transfer, $\tilde{t}=t-t^{min}_S$, so that a pattern does not depend on the calendar date of the episode. For account v we write T^{in}_v and T^{out}_v for its arrival and departure times and τ_v for the median of $T^{in}_v \cup T^{out}_v$. Each subgraph carries a label $y_S \in \{0,1\}$, and the goal is a scoring function $f(S) \in [0,1]$ that ranks laundering subgraphs above licit ones. Because laundering is the minority class, we evaluate with the illicit-class F_1 and the area under the precision–recall curve (PR-AUC).

B. Multi-Scale Timing in Real Ethereum Flows

We analysed the public Ethereum phishing data of Wu *et al.* [7], which contains the full transaction histories of 1,259 labelled phishing accounts between November 2016 and October 2018 (79,161 successful transfers: 48,601 incoming, 30,560 outgoing). Fig. 2 plots three timing distributions on a logarithmic axis. Gaps between consecutive events range from about one second to about one year, i.e., over seven orders of magnitude. The medians are 27 min between deposits, 9 min between outgoing transfers and 3.7 h for the residence time of value (deposit to next transfer out), but each distribution is broad: 27.2% of the inter-arrival gaps are shorter than 5 min while 6.7% fall between one day and one week, and the residence times spread over every band from minutes (10.3%) to more than one week (17.0%). No single time scale describes this behaviour, and a temporal encoding tuned to one band would compress the others. We therefore sample time on a geometric grid of $S=6$ scales, $\sigma_s \in$ {5 min, 30 min, 3 h, 18 h, 4.5 d, 27 d}, which are marked in Fig. 2.

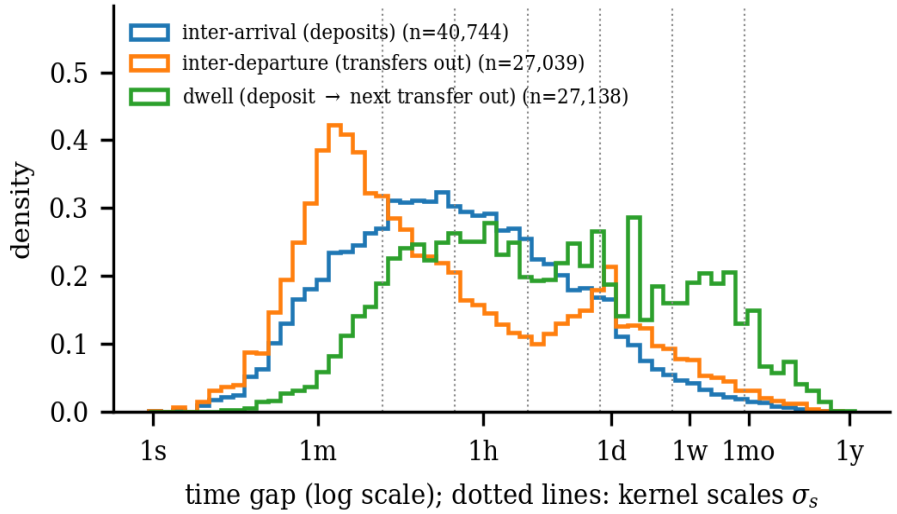


Figure 2Timing of 79,161 real transfers of 1,259 Ethereum phishing accounts (data of [7]). Inter-arrival, inter-departure and residence (dwell) times span seconds to months; dotted lines mark the six kernel scales of MS-TGT.

IV. The MS-TGT Model

Fig. 1 gives an overview. MS-TGT inherits the backbone of STEMP [17]: node tokens built from attributes, a functional time encoding and random-walk structural encodings; directed message passing with reverse messages [11]; a stack of transformer layers whose logits carry a shortest-path bias; a virtual readout; and a focal loss for class imbalance. We change three components, described in Sections IV-A to IV-C.

A. Multi-Scale Flow Spectrum Tokens

Let $\kappa_s(\Delta)=e^{-\Delta/\sigma_s}$ for $s=1,\ldots,S$. For an outgoing transfer of account v at time t, the residence time $\rho_v(t)=t-\max\{t' \in T^{in}_v : t'<t\}$ is the time that has passed since the latest arrival (infinite if value was never received before). The dwell spectrum of v is

$$D_{v,s} = \frac{1}{|T_v^{out}|} \sum_{t \in T_v^{out}} \kappa_s(\rho_v(t)), \; s = 1, \ldots, S. \quad (1)$$

$D_{v,s}$ is the Laplace transform of the residence-time distribution evaluated at frequency $1/\sigma_s$, scaled by the fraction of departures that follow an arrival. A pass-through node of a fast peeling chain has $D_{v,1}$ close to one, a mule that holds value for a few hours activates $D_{v,3}$ and $D_{v,4}$ but not $D_{v,1}$, and a dormant source activates only the slowest scales. Differences $D_{v,s+1}$-$D_{v,s}$ measure the probability mass of residence times around scale σ_s, in the same way that detail coefficients of a multi-resolution decomposition isolate one band [27]. Analogous spectra are computed for the gaps g between consecutive arrivals, $A_{v,s}$=mean$_g$\,$\kappa_s(g)$, and between consecutive departures, $O_{v,s}$, which quantify synchrony at each scale, and for dormancy, $M_{v,s}=\kappa_s(g^{max}{}_v)$ with $g^{max}{}_v$ the longest silent period of v. Each kernel acts as a soft indicator of "within σ_s", so $A_{v,s}$ is a soft count of near-simultaneous arrivals, i.e., of the fan-in temporal motifs of [18] evaluated at six windows. With two binary flags (more than one event; both arrivals and departures), $\psi_v \in \mathbb{R}^{4S+2}$ and the initial token is

$$z_v^{(0)} = W_f x_v + W_t \varphi(\backslash u\tilde{\tau}_v) + W_p p_v + W_\psi \psi_v, \quad (2)$$

where x_v are account attributes, φ is the learnable cosine time encoding of [12] (as used in STEMP) applied to $\tilde{\tau}_v=\tau_v-t^{min}{}_S$, and p_v are random-walk return probabilities. Computing ψ_v requires sorting the incident events of each node, $O(|E_S|\log|E_S|)$ per subgraph, and is done once in preprocessing. Two directed message-passing layers with in- and out-neighbour messages and edge features (log value, log relative time) follow, as in [11] and STEMP.

B. Multi-Scale Signed Temporal Attention Bias

Let $\Delta_{ij}=\tau_j-\tau_i$ be the signed time offset between nodes i and j and $\delta(i,j)$ their shortest-path distance. For head h, MS-TGT uses the attention logit

$$\alpha_{ij}^{(h)} = \frac{(W_Q z_i)^\top W_K z_j}{\sqrt{d_k}} + b_{\delta(i,j)}^{(h)} + \sum_{s=1}^{S} \beta_{s,\pm}^{(h)} \kappa_s(|\Delta_{ij}|), \quad (3)$$

where $\beta^{(h)}{}_{s,+}$ is used when j is later than i ($\Delta_{ij}>0$), $\beta^{(h)}{}_{s,-}$ when it is earlier, and their mean when $\Delta_{ij}=0$. The first two terms follow STEMP and Graphormer [14]; the third replaces STEMP's monotone function $g(|\Delta_{ij}|)$. The kernel bank is fixed and shared, so the bias adds only $2S$ parameters per head. We initialize every head with unit weight on one "home" scale (5 min, 3 h, 18 h and 27 d for the four heads), which gives scale-specialized heads at the start of training; all weights are then learned without constraints. The following properties make the difference to a monotone bias precise.

Proposition 1 (band-pass and causal selectivity). *(a)* If g is monotone, then $g(\Delta_2) \le \max\{g(\Delta_1), g(\Delta_3)\}$ for all $0<\Delta_1<\Delta_2<\Delta_3$: a monotone time-gap bias never prefers an intermediate lag. *(b)* For $\sigma_a<\sigma_b$, the member $f=\kappa_b-\kappa_a$ of the family in (3) satisfies $f(0)=0$, $f(\Delta)>0$ for $\Delta>0$, and has its unique maximum at $\Delta^*=\sigma_a\sigma_b \ln(\sigma_b/\sigma_a)/(\sigma_b-\sigma_a)$, e.g., 6.5 h for the pair (3 h, 18 h) and 1.6 d for (18 h, 4.5 d). *(c)* Any bias that depends on $|\Delta_{ij}|$ only assigns equal logits to a partner that acted Δ before node i and one that acted Δ after it; (3) separates the two whenever $\beta^{(h)}{}_{s,+} \neq \beta^{(h)}{}_{s,-}$ for some s.

Proof sketch. (a) follows from monotonicity. For (b), $f'(\Delta)=e^{-\Delta/\sigma a}/\sigma_a - e^{-\Delta/\sigma b}/\sigma_b$ is positive near zero and has a single root at Δ^*. (c) is immediate from the sign-dependent weights. □

Band-pass preferences matter for scheduled layering, where hops are separated by roughly one day and both near-simultaneous and month-apart partners are irrelevant, and the sign matters because value can only leave an account after it has arrived. Setting $\beta_+=\beta_-$ recovers a symmetric multi-scale bias and keeping a single non-zero scale recovers a single-scale exponential decay, so both are special cases that we evaluate as ablations.

C. Scale-Specific Readout and Gate

Instead of one virtual readout token, MS-TGT appends $R=3$ tokens, one per temporal band: fast (σ_1,σ_2; up to about half an hour), medium (σ_3,σ_4; hours to a day) and slow (σ_5,σ_6; days to

weeks). The attention logit from readout token r to node j receives the bias $u^{\top}_{h,r}\psi^{[r]}_{j}$, where $\psi^{[r]}_{j}$ collects the eight spectrum entries of band r, so each token is steered towards nodes whose flow behaviour is salient in its band. After L layers the readout states z_r are fused by a gate,

$$g_r = \frac{exp(w^{\top} tanh(W_g z_r))}{\sum_{r'} exp(w^{\top} tanh(W_g z_{r'}))}, z_S = \sum_{r=1}^{R} g_r z_r, \quad (4)$$

and the score is $\hat{y}_S=\sigma(\mathrm{MLP}(z_S))$. The model is trained with the focal loss [28], as in STEMP,

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} a_{y_i}(1-p_i)^{\gamma} \log p_i, \quad (5)$$

where p_i is the predicted probability of the true class, $\gamma=2$ and $a_1=0.75$. The gate vector g is part of the decision rule and reports whether a subgraph was judged on fast, medium or slow evidence. We did not include STEMP's prototype head in MS-TGT, because the scale gate already provides an inexpensive explanation; the two mechanisms are complementary and can be combined.

D. Complexity

For a subgraph with n nodes and m transfers, preprocessing costs $O(m\log m)$ for the spectrum and $O(n(n+m))$ for shortest paths. The kernel bias costs $O(HSn^2)$ per layer and is dominated by the $O(Hn^2d_k)$ attention itself, so MS-TGT has the same asymptotic cost as STEMP. With $d=32$, four heads and two layers it has 42.7K parameters, against 41.7K for our STEMP implementation.

V. Experimental Setup

A. TempoLaunder Benchmark

Public AML datasets either provide coarse time steps without edge timestamps (Elliptic has 49 steps about two weeks apart [2]) or subgraph labels without motif-level ground truth [4], and the large synthetic AMLworld data [20] targets transaction-level labels in a banking setting. To measure temporal reasoning at known scales we built TempoLaunder, a generator of blockchain-style candidate subgraphs in the spirit of the injected-motif positive control used to validate STEMP. Each sample consists of a licit background host and one injected motif. The host has 10–18 accounts (users, up to two exchange-like hubs and possibly a merchant) that are active during a 20–60-day window. Account activity follows a renewal process whose inter-event gaps are resampled from the 40,744 real Ethereum inter-arrival gaps of Section III-B (3.6 s to 60 days), and the host also contains licit structured behaviour: hot-wallet forwarding within minutes, synchronized sweeps of deposit addresses, payroll bursts, and daily or weekly recurring payments. Fast forwarding, synchrony and periodicity therefore also occur in licit activity and are not suspicious on their own.

The injected motif belongs to one of four families (Table I). Within a family, the laundering variant and two to four licit look-alikes share topology and amount distributions and differ only in their temporal signature: the time scale of the hops, synchrony, regularity, temporal order, or the combination of a slow and a fast scale. A quarter of the samples carry a laundering motif. We drop 3% of the transfers to mimic unobserved edges, add sub-minute block-time noise and remove isolated accounts; subgraphs contain 20 accounts and 45 transfers on average (at most 32 and 83). Training, validation and test sets contain 6,000, 1,500 and 3,000 subgraphs generated with independent seeds.

Table 1MOTIF FAMILIES OF TEMPOLAUNDER

Family	Laundering variant (label 1)	Licit look-alikes (label 0)
Chain: 4–6 hops with peel outputs	hops ≈18 min apart, time-respecting, after 5–25 d of source dormancy	≈20 h per hop, no dormancy; fast hops in scrambled order after dormancy; dormancy then ≈20 h hops; fast ordered hops without dormancy
Fan: 4–7 mules	fan-out within 15 min, holding ≈8 h, gather within 20 min	fan-out burst, gather spread over 1–20 d (payroll); fan-out over 15 d, synchronized sweep days later; burst and ≈8 h holding, unsynchronized gather
Layer: 2–3 branches of 2–3 hops	regular daily drip, period 16–32 h (±5%)	same median delay, irregular (log-normal, σ = 0.9); regular weekly drip (5–8 d)
Cycle: 3–5 hops back to origin	time-respecting, ≈2 h per hop, 85–95% returned	≈4 d per hop, 100–110% returned (loan); fast, scrambled order; ≈4 d per hop, 85–95% returned

Shifted test sets. To emulate launderers that adapt their timing, a threat also discussed for STEMP, we regenerate the test set with the same random seed, so that hosts and motifs stay paired with the i.i.d. test set, but change the laundering timing: (i) all laundering delays are slowed down by a factor of three, (ii) an order-preserving delay drawn from *U(0,1* h*)* is added to every laundering hop, and (iii) the same with *U(0,4* h*)*. Licit look-alikes are unchanged and no model is retrained.

B. Real-Data Temporal Probe

The Ethereum data of [7] contain complete timestamped histories only for phishing accounts, so we use them to probe temporal perception rather than detection. We cut the history of each phishing account into windows of consecutive transfers (at most 64 transfers and 31 counterparties, at least 12 transfers spanning at least two days) and keep up to three windows per account, which yields 1,133 star-shaped transaction subgraphs from 816 accounts. For each real window we draw one randomized reference model [19] that keeps topology, directions and values but destroys temporal organisation at a chosen scale. *Shuffle* permutes timestamps across transfers. *Fine* adds independent offsets drawn from *U(-1* h*,1* h*)* and erases sub-hour structure only. *Coarse* permutes the occupied day-blocks while keeping intra-day offsets, which erases the order of days but preserves fine structure. Each task asks a model to separate real windows from their randomized counterparts (balanced classes); accounts are split 60/20/20 into training, validation and test sets.

C. Baselines and Training Protocol

We compare MS-TGT with ten baselines. XGBoost [29] receives 45 hand-crafted subgraph descriptors: size, degree and amount statistics, quantiles of inter-event gaps, the fractions of residence times and of arrival and departure gaps below 5 min, 1 h, 1 d and 1 w, maximum silent periods and the size of strongly connected components; it therefore sees multi-window temporal statistics at the subgraph level. The message-passing baselines are GCN [8], GIN [10], GAT [9] and a Multi-GNN-style directed model with reverse messages and edge features [11]. The graph transformers are Graphormer [14], with shortest-path bias and a virtual readout, and GraphGPS [15], with random-walk encodings. TGAT [12] attends over temporal neighbourhoods with functional encodings of time differences, followed by pooling. STEMP [17] is re-implemented from its Eqs. (2)–(17): tokens with time and random-walk encodings, a directed pre-encoder, shortest-path and monotone 32-bucket time-

gap biases, a virtual readout, 16 motif prototypes (12 illicit, 4 licit) fused with the subgraph embedding, the clustering, separation and diversity objectives with the published weights, and periodic prototype projection; the Elliptic1 pre-training stage does not apply to our data and is omitted. In pilot runs the published prototype-loss weights lowered the validation PR-AUC in our setting (0.80 against 0.95 without the auxiliary objectives), so we also report STEMP-core, which keeps the prototype head but disables the auxiliary objectives. It is the stronger of the two variants and serves as the main reference for MS-TGT. All models receive the same node attributes, including temporal summaries (first, last and median activity time, activity span), so that no baseline is deprived of time information; the flow spectrum is the only input reserved for MS-TGT, and its value for other models is tested separately. Following STEMP's fairness protocol, every neural model uses width $d=32$, four attention heads, two transformer layers (three layers for message-passing baselines), AdamW with learning rate 10^{-3} and weight decay 10^{-4}, batch size 64, at most 12 epochs with early stopping on validation PR-AUC (patience 4), the focal loss with $\gamma=2$ and $a_1=0.75$, and a decision threshold that maximizes the validation F_1 and is applied unchanged to all test sets. Apart from the STEMP loss weights, no model received individual tuning. All models were implemented in JAX and trained on a single CPU core (Intel Xeon, 2.3 GHz).

D. Metrics and Statistical Protocol

We report illicit-class F_1, precision, recall, PR-AUC, ROC-AUC and the Matthews correlation coefficient (MCC) as mean ± standard deviation over five seeds, which control initialization and data order; the data splits are fixed. An exact paired Wilcoxon signed-rank test over five seeds cannot reach $p<0.05$ (its smallest two-sided p-value is 0.0625), so we follow the evaluation protocol of STEMP: exact two-sided Wilcoxon signed-rank tests over the paired seeds (reported for completeness), instance-level bootstrap confidence intervals of the seed-averaged F_1 difference (10,000 resamples of the 3,000 test subgraphs), McNemar tests on paired decisions, and Benjamini–Hochberg control of the false-discovery rate across baselines, in line with the recommendations of [30]

Table 2RESULTS ON THE TEMPOLAUNDER I.I.D. TEST SET (MEAN ± STD OVER FIVE SEEDS; THE LAST FOUR COLUMNS GIVE THE ILLICIT-CLASS F1 PER MOTIF FAMILY)

Model	F_1	Precision	Recall	PR-AUC	ROC-AUC	MCC	Chain	Fan	Layer	Cycle
XGBoost [29]	0.581 ±0.003	0.484±0.025	0.735±0.050	0.637±0.004	0.830±0.002	0.426 ±0.006	0.521	0.773	0.530	0.512
GCN [8]	0.654 ±0.017	0.520±0.023	0.882±0.014	0.659±0.037	0.881±0.012	0.539 ±0.023	0.564	0.667	0.758	0.657
GIN [10]	0.690 ±0.012	0.563±0.026	0.895±0.028	0.752±0.022	0.908±0.008	0.590 ±0.012	0.580	0.722	0.832	0.659
GAT [9]	0.687 ±0.016	0.569±0.016	0.869±0.053	0.760±0.013	0.910±0.006	0.583 ±0.026	0.571	0.735	0.823	0.649
Multi-GNN-style [11]	0.800 ±0.016	0.699±0.021	0.934±0.023	0.888±0.026	0.962±0.008	0.737 ±0.022	0.683	0.913	0.936	0.711
Graphormer [14]	0.736 ±0.014	0.660±0.022	0.835±0.038	0.827±0.010	0.936±0.005	0.647 ±0.019	0.641	0.802	0.881	0.657

GraphGPS [15]	0.749 ±0.02 2	0.649±0 .028	0.88 8±0. 039	0.833±0. 034	0.941±0 .012	0.667 ±0.03 1	0.636	0.843	0.887	0.668
TGAT [12]	0.783 ±0.01 4	0.693±0 .028	0.90 1±0. 020	0.873±0. 019	0.956±0 .007	0.712 ±0.01 8	0.660	0.893	0.900	0.711
STEMP [17]	0.730 ±0.04 5	0.649±0 .040	0.83 7±0. 065	0.816±0. 049	0.931±0 .019	0.638 ±0.06 4	0.607	0.855	0.862	0.645
STEMP-core	0.815 ±0.00 8	0.721±0 .019	0.93 9±0. 031	0.908±0. 011	0.969±0 .004	0.757 ±0.01 2	0.676	0.995	0.937	0.708
MS-TGT (ours)	0.949 ±0.00 4	0.925±0 .012	0.97 4±0. 009	0.983±0. 002	0.995±0 .000	0.933 ±0.00 5	0.934	0.980	0.959	0.924

VI. Results

A. Main Comparison

Table II summarizes the i.i.d. test results. MS-TGT reaches an illicit-class F_1 of 0.949±0.004 and a PR-AUC of 0.983±0.002. The strongest baseline is STEMP-core (0.815±0.008 and 0.908±0.011), followed by the Multi-GNN-style directed model (0.800) and TGAT (0.783). Most of the improvement comes from precision (0.721 to 0.925), i.e., from fewer licit look-alikes being flagged, while recall also rises (0.939 to 0.974). The ranking of the baselines follows the temporal and directional information they can use: undirected message passing with node-level time summaries (GCN, GIN, GAT) reaches 0.654–0.690, global attention without time (Graphormer, GraphGPS) 0.736–0.749, and models that see edge directions or edge times 0.783–0.815. XGBoost with subgraph-level multi-window statistics reaches only 0.581: the fractions of fast forwards or synchronized arrivals are diluted by licit background activity with the same local statistics. STEMP with the published prototype-loss weights obtains 0.730±0.045; disabling the auxiliary objectives raises its F_1 by 8.4 points and lowers its variance, which supports the use of STEMP-core as the reference.

Statistical significance. Against STEMP-core, the mean paired F_1 difference over the five seeds is 0.134 (95% t-interval [0.121, 0.148]) and all five seeds favour MS-TGT. With five pairs the exact Wilcoxon test cannot go below $p=0.0625$ (Section V-D), so significance rests on instance-level tests: the bootstrap 95% interval of the seed-averaged difference is [0.120, 0.150] with no non-positive resample among 10,000 ($p<2\times10^{-4}$), and McNemar's test on paired decisions gives a median of 261 subgraphs corrected and 35 broken by MS-TGT per seed ($p<10^{-34}$ in every seed). The same holds for the other nine baselines, and all differences remain significant after Benjamini–Hochberg correction.

B. Which Patterns Benefit

The per-family columns of Table II locate the gain. On chains and cycles, where laundering differs from its look-alikes by the conjunction of a fast scale and temporal order (plus a slow dormancy scale for chains), MS-TGT improves the F_1 from at most 0.683 and 0.711 for the best baseline to 0.934 and 0.924. These are the families that the residence-time spectrum encodes directly: a pass-through node of a fast time-respecting chain responds strongly at short scales, a node of a scrambled chain hardly responds because many departures precede the arrival, and a dormant source responds only at slow scales. On layering, MS-TGT is ahead by 2.2 points. On the fan family STEMP-core is better (0.995 against 0.980):

synchronized fan-out and gather give the mules nearly identical representative times, which STEMP's time-gap bias captures directly.

C. Robustness to Temporal Evasion

Fig. 3 reports the F_1 on the paired shifted test sets. Delaying every laundering hop by up to one hour barely affects MS-TGT (0.945), whereas the best baseline, STEMP-core, falls to 0.786. With up to four hours of delay MS-TGT keeps 0.886, against 0.678 for the best baseline (TGAT) and 0.601 for STEMP-core. Slowing all laundering delays by a factor of three is the hardest shift because it moves laundering hops towards the timescales of their look-alikes: MS-TGT drops to 0.765, which is still 20.3 points above the best baseline (STEMP-core, 0.562), while the Multi-GNN-style model collapses from 0.800 to 0.441. Spreading the temporal evidence over six scales thus degrades gracefully when an adversary changes the timescale, but it does not make the model invariant to such changes. Cost-sensitive adversarial training for cloud intrusion detection demonstrates an effective experimental perspective on the trade-off between adversarial robustness and asymmetric detection costs [32].

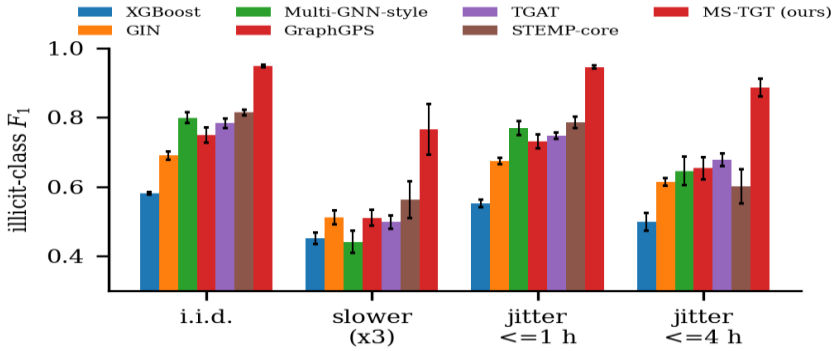


Figure 3Illicit-class F1 on the i.i.d. test set and on the paired shifted test sets (laundering delays slowed by a factor of three, or delayed by up to 1 h or 4 h per hop); mean and standard deviation over five seeds

D. Ablation and Plug-In Analysis

Table 3ABLATION OF MS-TGT (SEED 0, TOP) AND PLUG-IN OF THE FLOW SPECTRUM (MEAN ± STD OVER SEEDS 0–2, BOTTOM)

Variant	F_1	PR-AUC	F_1 slowed	F_1 4-h jitter
MS-TGT (full)	0.950	0.986	0.735	0.889
w/o flow spectrum ψ	0.905	0.960	0.688	0.811
monotone bias (as STEMP)	0.952	0.983	0.753	0.790
symmetric multi-scale bias	0.957	0.987	0.747	0.869
single readout token	0.949	0.980	0.673	0.876
GIN [10]	0.690± 0.015	0.754± 0.027	0.506± 0.021	0.617± 0.009
GIN + ψ	0.875± 0.006	0.941± 0.002	0.652± 0.023	0.680± 0.013
STEMP-core	0.811± 0.007	0.903± 0.011	0.558± 0.068	0.577± 0.050
MS-TGT	0.949± 0.003	0.983± 0.002	0.768± 0.056	0.878± 0.026

Table III removes one component at a time and adds the flow spectrum to a baseline. Removing the spectrum costs 4.5 points of F_1 on the i.i.d. test set (0.950 to 0.905) and 7.9

points under the 4-h jitter, the largest effect of any single component. Replacing the multi-scale signed bias by STEMP's monotone bias, making it symmetric, or using a single readout token changes the i.i.d. F_1 by less than one point. On the shifted test sets the picture is mixed: the largest loss is 10.0 points for the monotone bias on the 4-h jitter test set, and the largest change in the other direction is +1.8 points for the monotone bias on the slowed test set. Without the multi-scale kernels (monotone bias) the F_1 under the 4-h jitter drops by 10.0 points, and with a single readout token the F_1 on the slowed set drops by 6.1 points, so on this seed the kernel bias mainly protects against delayed hops and the scale-specific readout against a slower timescale; with a single seed these effects still need confirmation. The spectrum is also useful outside MS-TGT: added to the node features of GIN it raises the F_1 from 0.690 to 0.875, more than any architectural change among the baselines. We therefore regard the flow spectrum as the main source of i.i.d. accuracy, and the attention-level components as contributing robustness under temporal shift together with a readable temporal focus.

E. Real-Data Temporal Probe

Table 4REAL-DATA PROBE: ROC-AUC FOR SEPARATING REAL ETHEREUM PHISHING WINDOWS FROM RANDOMIZED REFERENCE MODELS (MEAN ± STD OVER TWO SEEDS)

Model	Shuffle	Fine (±1 h)	Coarse (days)
XGBoost	0.908±0.000	0.798±0.006	0.744±0.000
GIN	0.803±0.020	0.507±0.003	0.740±0.015
STEMP-core	0.839±0.013	0.595±0.008	0.751±0.011
MS-TGT	0.874±0.010	0.639±0.007	0.749±0.006

Table IV lists the probe results on the real Ethereum windows. All models detect the full timestamp shuffle; XGBoost is best (0.908), followed by MS-TGT (0.874), STEMP-core (0.839) and GIN (0.803). The fine-scale task separates the models: GIN, which only sees node-level time summaries, is at chance (0.507), STEMP-core reaches 0.595 and MS-TGT 0.639, whereas XGBoost reaches 0.798 through its explicit fraction of gaps below five minutes. On the coarse task all models perform alike (0.740–0.751). In these star-shaped windows the evidence for sub-hour organisation is essentially a global gap statistic, which an explicit histogram learns from 1,368 training windows more efficiently than an attention model. The flow spectrum still lifts the transformer above STEMP-core on the shuffle and fine tasks, but the probe shows that the large margins of Table II depend on topology–time coupling and do not transfer automatically to every real setting.

F. Interpretability and Efficiency

Fig. 4(a) averages the scale gate over the test set. For licit look-alikes it is close to uniform (0.30–0.35 per band). For laundering chains and cycles the medium band receives 0.48 and 0.47 of the weight, while fan and layer laundering stay near uniform. The medium band is not where chain hops occur (minutes) but the band whose kernels best separate them from their look-alikes: the 3-h kernel maps an 18-min hop to 0.90 and a 20-h hop to 0.001. The gate thus points an analyst to the discriminating time scale, which complements the motif-level explanations of STEMP's prototypes. Fig. 4(b) shows the learned bias profiles of the seed-0 model: 2 of the four heads have their forward maximum at an intermediate lag (13

min, 6.3 h), i.e., a band-pass profile that a monotone bias cannot express, and forward and backward profiles differ by up to 0.95 logits. MS-TGT has 42.7K parameters, needs 5.9 s per training epoch on one CPU core (STEMP: 6.4 s, STEMP-core: 5.7 s) and scores 7,375 subgraphs per second, against 8,541 for STEMP-core; the spectrum is computed once per subgraph during preprocessing.

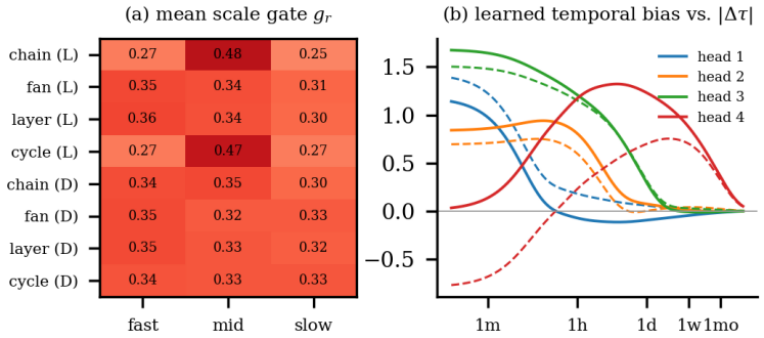


Figure 4 (a) Mean scale gate of MS-TGT per motif family for laundering (L) and look-alike (D) test subgraphs (five seeds). (b) Learned temporal bias of the four heads of the seed-0 model as a function of the time offset; solid: later partner (Δ>0), dashed: earli

VII. Discussion and Limitations

Three limitations qualify these results. First, TempoLaunder is a controlled benchmark. Its background timing is resampled from real transfers and every laundering motif has licit look-alikes, but the laundering signatures were specified by us, and the flow spectrum measures quantities (residence time, synchrony, dormancy) that coincide with the generator's design dimensions as well as with common AML red flags. The margins in Table II therefore show that multi-scale flow descriptors are learnable and useful; they are not an estimate of field performance, for which the real-data probe, with smaller margins and a strong gradient-boosting baseline, is the more conservative indicator. Second, the attention-level components changed the i.i.d. accuracy by less than one point in our single-seed ablation and showed their benefit only under temporal shift and in interpretability, and a multi-seed factorial ablation that also equips STEMP-core with the flow spectrum is needed to separate input and architecture effects; whether band-pass and causal biases help on real laundering cases with less clean signatures remains open. Third, the experiments use small models on a single CPU core, five seeds for the main comparison and simple, non-adaptive evasions; an adversary who knows the kernel grid could place delays between scales, which motivates learning the scales and an adaptive adversarial evaluation. Labelled blockchain data with fine-grained timestamps would allow these questions to be answered on real cases.

VIII. Conclusion

We presented MS-TGT, a multi-scale temporal graph transformer for subgraph-level blockchain AML that extends the time-biased subgraph transformer STEMP with a multi-scale flow spectrum, a signed mixture-of-kernels attention bias and scale-specific readout tokens. Real Ethereum transfers show residence times spread over seven orders of magnitude. On a benchmark calibrated with these timings, MS-TGT improved the illicit-class F_1 from 0.815 to 0.949 over the strongest of ten baselines and degraded least under temporal evasion. The flow spectrum carries most of the gain and also improves other architectures, and the scale gate identifies the time band that separates laundering from look-alike behaviour. Future work will combine the spectrum with STEMP's motif prototypes and cross-chain pre-training, learn the kernel scales, and validate the approach on labelled real-world data.

REFERENCES

S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, "Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics," in *Proc. KDD Workshop Anomaly Detection in Finance*, 2019, arXiv:1908.02591.

Y. Elmougy and L. Liu, "Demystifying fraudulent transactions and illicit nodes in the Bitcoin network for financial forensics," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discov. Data Mining (KDD)*, 2023, pp. 3979–3990, doi: 10.1145/3580305.3599803.

C. Bellei, M. Xu, R. Phillips, T. Robinson, M. Weber, T. Kaler, C. E. Leiserson, Arvind, and J. Chen, "The shape of money laundering: Subgraph representation learning on the blockchain with the Elliptic2 dataset," 2024, arXiv:2404.19109.

S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A fistful of bitcoins: Characterizing payments among men with no names," in *Proc. Internet Meas. Conf. (IMC)*, 2013, pp. 127–140, doi: 10.1145/2504730.2504747.

M. Möser, R. Böhme, and D. Breuker, "An inquiry into money laundering tools in the Bitcoin ecosystem," in *Proc. APWG eCrime Researchers Summit*, 2013, pp. 1–14, doi: 10.1109/eCRS.2013.6805780.

F. Zhou, "Moment-constrained side-force allocation for wind-assisted merchant ships: Hydrodynamic performance of retractable anti-leeway fins and flap rudders," Innovation and Technology Studies, vol. 3, no. 1, pp. 27–41, 2026.

T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.

J. Chen, X. Zhang, and X. Han, "Explainable identification of tax-related financial reporting anomalies in U.S. public companies: Cross-fitted tax-consistency residuals and Owen-value Shapley propagation," World Journal of Economics and Business Research, vol. 1, no. 1, pp. 33–45, 2023.

D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, "Inductive representation learning on temporal graphs," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, "Temporal graph networks for deep learning on dynamic graphs," 2020, arXiv:2006.10637.

C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu, "Do transformers really perform badly for graph representation?" in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, 2021, pp. 28877–28888.

L. Rampášek, M. Galkin, V. P. Dwivedi, A. T. Luu, G. Wolf, and D. Beaini, "Recipe for a general, powerful, scalable graph transformer," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, 2022, pp. 14501–14515.

U. Alon and E. Yahav, "On the bottleneck of graph neural networks and its practical implications," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

Y. Lin, X. Yue, M. Bian, R. Zhu, and D. Teng, "Subgraph-level temporal graph transformer with laundering-motif prototypes for anti-money laundering on blockchains," *IEEE Access*, vol. 14, pp. 139135–139159, 2026, doi: 10.1109/ACCESS.2026.3731160.

A. Paranjape, A. R. Benson, and J. Leskovec, "Motifs in temporal networks," in *Proc. 10th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2017, pp. 601–610, doi: 10.1145/3018661.3018731.

L. Gauvin, M. Génois, M. Karsai, M. Kivelä, T. Takaguchi, E. Valdano, and C. L. Vestergaard, "Randomized reference models for temporal networks," *SIAM Rev.*, vol. 64, no. 4, pp. 763–830, 2022, doi: 10.1137/19M1242252.

E. Altman, J. Blanuša, L. von Niederhäusern, B. Egressy, A. Anghel, and K. Atasu, "Realistic synthetic financial transactions for anti-money laundering models," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), Datasets and Benchmarks Track*, vol. 36, 2023.

X. Li, S. Liu, Z. Li, X. Han, C. Shi, B. Hooi, H. Huang, and X. Cheng, "FlowScope: Spotting money laundering based on graphs," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 4731–4738, doi: 10.1609/aaai.v34i04.5906.

A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. B. Schardl, and C. E. Leiserson, "EvolveGCN: Evolving graph convolutional networks for dynamic graphs," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 5363–5370, doi: 10.1609/aaai.v34i04.5984.

J. Chen, X. Han, X. Zhang, and Y. Yang, "Exact concept-level explanations for early detection of financial misstatements," Eurasia Journal of Science and Technology, vol. 5, no. 2, pp. 34–48, 2023.

P. Holme and J. Saramäki, "Temporal networks," *Phys. Rep.*, vol. 519, no. 3, pp. 97–125, 2012, doi: 10.1016/j.physrep.2012.03.001.

A.-L. Barabási, "The origin of bursts and heavy tails in human dynamics," *Nature*, vol. 435, no. 7039, pp. 207–211, 2005, doi: 10.1038/nature03459.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 5998–6008.

S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989, doi: 10.1109/34.192463.

T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2999–3007, doi: 10.1109/ICCV.2017.324.

T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD)*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

X. Yue and R. Zhu, "An explainable hybrid machine learning framework for cash-flow-conditioned multi-horizon liquidity risk early warning: An SME-oriented benchmark study," World Journal of Management Science, vol. 2, no. 1, pp. 63–72, 2024.

J. Ding et al., "Game-Theoretic Cost-Sensitive Adversarial Training for Robust Cloud Intrusion Detection Against GAN-Based Evasion Attacks," Applied Sciences, vol. 16, art. 3944, 2026, doi: 10.3390/app16083944.