TempoTriads: Streaming Estimation of Temporal Triadic Motifs for Social-Computing Streams

Aleksandar Stanković[1 (✉)] and Haoran Du[2]

[1] Faculty of Technical Sciences, University of Novi Sad,
Novi Sad 21000, Serbia
stankovic.sv25.2022@uns.ac.rs

[2] Shanghai Key Lab of Intelligent Information Processing,
School of Computer Science, Fudan University,
Shanghai 200433, China
hrdu24@m.fudan.edu.cn

Abstract. Social platforms and organizational communication streams exhibit higher-order interaction patterns—triadic closures—that reflect key social mechanisms such as reciprocity, transitivity, and conversational cascades. We study one-pass estimation of *temporal* triadic closures within a sliding window Δ and introduce TempoTriads, a streaming estimator that maintains a compact reservoir of *active wedges* and applies Horvitz–Thompson weighting at closure time to produce *unbiased*, anytime totals with calibrated uncertainty (moving block bootstrap). TempoTriads supports directed/undirected graphs and a *typed* regime where all three edges share an event type (e.g., `reply`, `forward`, `trust`), enabling semantically coherent telemetry for governance and safety analytics. Across three public social interaction graphs (CollegeMsg, email-Eu, BitcoinOTC), TempoTriads achieves single-digit median relative error at practical memory budgets, with conservatively calibrated confidence intervals. We release an open-source implementation with one-command scripts for full reproduction and discuss privacy-preserving, aggregate-only deployments suitable for policy and auditing workflows.

Keywords: Temporal Motifs; Streaming; Social Networks; Governance Analytics; Privacy

1 Introduction

On high-velocity social platforms—such as Weibo/Reddit communities and enterprise IM—operations and governance teams often need minute-level telemetry on how three-person interaction structures evolve. Two patterns are often actionable in platform policy and incident-response playbooks: (i) the escalation of reciprocal ties (e.g., mutual replies/likes or trust building in peer trading) and (ii) short interaction cascades (e.g., `forward`→`reply`→`re-forward`) that can prefigure topic diffusion or coordinated behavior. When changes in such triadic structures are detected late, moderation and response costs typically rise;

in contrast, aggregate, privacy-preserving rates with uncertainty align better with policy dashboards than user-level logs.

Temporal motifs provide a principled lens on these signals and are known to vary by domain and timescale [15]. In many monitoring workflows, typed semantics (e.g., `reply`, `forward`, `trust`) are required so indicators map cleanly to governance taxonomies and interpretability pipelines [2]. However, obtaining timely triadic-closure rates under a sliding window Δ is challenging in streams: wedge populations can spike, closures and expirations interleave, and naïve window maintenance collides with memory and latency limits. Prior estimation frameworks for temporal motifs show that sampling must handle window/interval boundary effects to remain unbiased and low-variance [13]; complementary systems target specific motifs and windows (e.g., triangle counting under sliding windows or butterflies on temporal bipartite streams) with strong maintenance and scalability, but they typically emphasize exact or specialized counts rather than unbiased, anytime estimates with calibrated uncertainty [6,1,5].

We study one-pass estimation of temporal triadic closures within a sliding window Δ and introduce TempoTriads, a compact streaming estimator that maintains a small sample of active wedges and applies Horvitz–Thompson weighting at closure time. We support directed and undirected graphs and a typed-homogeneous regime (all three edges share an `etype`), producing semantically coherent telemetry for governance and safety analytics. Beyond unbiasedness, we deliver calibrated confidence intervals via a moving-block bootstrap to support risk- and policy-oriented decision-making. In the broader context of streaming graph analytics, we connect to fixed-memory, unbiased triangle estimation in fully dynamic streams (e.g., TRIÈST [18,19]) and recent scalable temporal motif systems (e.g., MoTTo [12]), while targeting a distinct object: *close-once temporal closure events* under a live sliding window with anytime HT totals.

Relevance to social computing. We frame triadic-closure telemetry as a governance-analytics primitive: minute-level rates for reciprocity and short cascades summarize conversational health and coordination *without* storing user-level logs. We argue these aggregate, uncertainty-aware signals align with policy dashboards and controlled interventions (e.g., onboarding tweaks, rate-limit changes), support fairness and auditing via typed breakdowns (`reply`/`forward`/`trust`), and complement qualitative moderation workflows in online communities and organizational communication streams.

Contributions. We (i) design a one-pass estimator for sliding-window triadic closures using a bottom-k reservoir of active wedges with Horvitz–Thompson weights for unbiased, anytime totals; (ii) quantify uncertainty via a moving-block bootstrap and stabilize variance by simple degree$\times$recency stratification; (iii) support typed-homogeneous triads on directed/undirected graphs to align counts with governance taxonomies; and (iv) release reproducible artifacts showing single-digit median relative error at modest budgets on three public graphs.

2 Related Work

Temporal motifs. Temporal motifs provide a structured lens on time-stamped interactions and have seen wide use in temporal network analysis [7,15].

Streaming motif estimation. Space-efficient estimators over streams include triangle/wedge sampling and temporal-motif samplers [9]. For triangle counting in fully dynamic streams, fixed-memory unbiased methods such as TRIÈST [18,19] established bottom-k/priority sampling as a practical backbone. Many approaches target different motif definitions (temporal paths vs. closures), assume offline passes/batches, or rely on heuristic inclusion rather than explicit probability accounting; in contrast, we compute Horvitz–Thompson (HT) weights at *closure time* under a live sliding window.

Sampling with HT weights and stratification. Bottom-k (a.k.a. priority/bottom-k) sampling with Horvitz–Thompson (HT) weighting is a standard route to unbiased estimation in streaming analytics; stratification further reduces variance by allocating samples to high-variance strata [3,4].

Uncertainty under dependence. For dependent sequences, block-resampling methods (moving/stationary block bootstrap) yield pragmatic confidence intervals without strong mixing assumptions [10,16].

Positioning and novelty. Recent systems report strong scaling for temporal motif counting or temporal-path estimators (e.g., MoTTo [12], TEACUPS [14]), while classical streaming methods address triangles under fully dynamic updates (e.g., TRIÈST [18,19]). *TempoTriads* differs along four axes: (i) the *target* is triadic *closure events* under a sliding window with close-once semantics; (ii) a compact *bottom-k reservoir over active wedges* yields explicit inclusion probabilities and *unbiased* HT totals (uniform and stratified); (iii) we support a *typed-homogeneous* regime (all three edges share `etype`); and (iv) we provide *calibrated* uncertainty via a moving-block bootstrap tailored to streaming dependence.

3 Preliminaries and Problem Definition

Temporal triads. We model a time-ordered stream of directed edges $(u \to v, t)$ and a sliding horizon Δ. We define a *wedge* as two edges sharing a center with times $t_1 < t_2$. A wedge *closes* when a third edge arrives at t_3 with $t_2 < t_3 \leq t_1 + \Delta$. In directed graphs, we group closures into three families: Cascade (two edges point toward the closer), Cycle3 (a 3-cycle), and RecipClosure (reciprocation followed by an outward edge). In undirected graphs we treat a single family, Undirected. We further distinguish between closures (closer in (t_1, t_2) for reciprocity) and after closures (closer in $(t_2, t_1+\Delta]$).

Typed setting. When edges carry an `etype`, we count *typed-homogeneous* closures in which all three edges share the same `etype`.

Target quantity. We estimate *closure events*: each wedge contributes at most once upon its first admissible closer (by family) within Δ. We denote totals by family Y_f and aggregate $Y_{\text{tot}} = \sum_f Y_f$.

Table 1. Notation: key symbols used throughout.

Symbol	Meaning
Δ	Sliding time horizon (window)
$(u \to v, t)$	Directed edge at timestamp t
Wedge	Two time-ordered edges sharing a center; $t_1 < t_2$
Active wedge	Formed but not yet closed or expired (deadline $t_1 + \Delta$)
A_t	# of active wedges at time t (population size)
B	Reservoir size (uniform); $B_s(t)$ for stratum s at time t
π_t	Inclusion probability at close time ($\approx B/A_t$)
$\hat{Y}_f$	HT estimate of total closures for family f
$\hat{Y}_{\text{tot}}$	HT estimate of aggregate closures $\sum_f \hat{Y}_f$

4 Method

Active-wedge reservoir. We let A_t denote the number of active wedges at time t (formed but not yet closed or expired). We assign each wedge an i.i.d. key $U \sim \text{Unif}(0,1)$ and retain the B smallest keys; we maintain the threshold $\tau_t = \max\{U \text{ in the reservoir}\}$. At a closer time t_3, if the wedge is in-reservoir we add an HT contribution $1/\pi_{t_3}$, where $\pi_{t_3} = \Pr(U \le \tau_{t_3}) \approx R_{t_3}/A_{t_3}$ and R_{t_3} is the number of active wedges with keys $\le \tau_{t_3}$. Bottom-k with i.i.d. uniform keys selects an equiprobable subset of the active population, so $\pi_{t_3} \approx \min\{1, B/A_{t_3}\}$ (or stratum-wise). We maintain per-family sums $\hat{Y}_f = \sum_{i \in \mathcal{C}_f} 1/\pi_i$ and the total $\hat{Y}_{\text{tot}} = \sum_f \hat{Y}_f$. We implement reservoir updates with standard reservoir/bottom-k techniques [20,3] and compute HT weights [8] at closure time.

Algorithm 1 TempoTriads: Per-edge update under sliding window Δ

Require: Incoming edge $e = (u \to v, t)$; window Δ; budget B (or per-stratum B_s).
  State: recent-neighbor deques $\mathcal{N}$; reservoir R (or R_s); threshold τ (or τ_s);
  active counts A_t (or $A_{s,t}$); family totals $\hat{Y}_f$.
Expire edges with time $< t - \Delta$ and wedges with deadline $t_1 + \Delta < t$;
  drop expired wedges from R (if present) and update A_t.
Close any wedge w that e completes (respecting family and typed-homogeneous constraints).
if $w \in R$ **then**
  Add $1/\pi_t$ to the corresponding $\hat{Y}_f$, where $\pi_t \approx |R|/A_t$ (or $|R_s|/A_{s,t}$).
end if
Mark w *closed* (close-once) and remove it from indexes; update A_t.
Form new wedges with e and compatible predecessor edges in $\mathcal{N}$;
  for each new wedge w' sample key $U \sim \text{Unif}(0,1)$ and set deadline $t_1 + \Delta$.
if $|R| < B$ **or** $U < \tau$ **then**
  Insert w' into R; if $|R| > B$, evict the item with the largest key and update τ.
end if
Enqueue e into $\mathcal{N}$; update diagnostics (inclusion rate, occupancy).

Uniform vs. stratified sampling. In uniform mode there is a single reservoir. In stratified mode we partition wedges by center degree buckets (quantiles, e.g., $\leq q_{0.5}, \leq q_{0.9}, > q_{0.9}$) and by the recency gap $t_2 - t_1$ (fractions of Δ, e.g., $\leq 0.33\Delta, \leq 0.66\Delta, > 0.66\Delta$). Each stratum s maintains its own reservoir and threshold, yielding stratum-specific $\pi_{s,t}$ and contributions $1/\pi_{s,t}$.

Typed support. In typed mode, wedge creation, waits, and closures are keyed by `etype`. A typed wedge can only be formed and closed by edges that share the same `etype`.

Uncertainty quantification. We aggregate contributions into fixed edge-count blocks and apply a moving block bootstrap: resample blocks with replacement, sum per replicate, and take percentile intervals for each family and the total.

Data structures and invariants. We index recent in/out neighbors per node using deques, track waits for closer pairs and reciprocity *after*, and lazily expire wedges at $t_1 + \Delta$. We enforce close-once semantics and remove wedges from all maps upon close/expire.

Complexity. Stream processing is linear in edges with $O(1)$ expected reservoir ops; memory is $O(B)$ plus small indexes per recent neighbor. Exact baselines use wedge-first enumeration with binary-searched closers.

4.1 Streaming estimator.

Let $\mathcal{C}$ denote the set of true closure events (by our close-once semantics) within the stream and window Δ. In uniform mode we maintain a bottom-k reservoir of size B over the *active* wedges; at close time t_3 the inclusion probability of the closed wedge is $\pi_{t_3} \approx B/A_{t_3}$ (or R_{t_3}/A_{t_3} when tracking the threshold explicitly). Each observed closure contributes $1/\pi_{t_3}$ to the appropriate family sum.

We emphasize that bottom-k with i.i.d. uniform keys yields an equiprobable subset of the active population at any time t, so the inclusion rate at closure is $\pi_t \approx \min\{1, B/A_t\}$ (or per stratum), which makes HT weighting immediate.

Proposition 1 (HT unbiasedness for temporal closure totals). *The Horvitz–Thompson estimators $\hat{Y}_f = \sum_{c \in \mathcal{C}_f} I_c/\pi_c$ and $\hat{Y}_{\text{tot}} = \sum_f \hat{Y}_f$ are unbiased for the corresponding population totals of closure events, where I_c is the indicator that the wedge underlying closure c is included at the close time and π_c is its inclusion probability under the reservoir scheme in effect at that time.*

Proof (Proof sketch). Fix a close time t and the realized active-wedge set W_t of size A_t (or per stratum $A_{s,t}$) and fixed budgets B (or $B_s(t)$). With i.i.d. Uniform(0,1) keys, bottom-k selects a simple random subset of size $\min\{B, A_t\}$, so for any closure $c \in W_t$, $\Pr(I_c = 1 \mid W_t) = \min\{1, B/A_t\}$ (resp. $B_s(t)/A_{s,t}$). Hence $\mathbb{E}[I_c/\pi_c \mid W_t] = 1$ and, by linearity, $\mathbb{E}[\sum_{c \in \mathcal{C}_f} I_c/\pi_c \mid W_t] = |\mathcal{C}_f|$. Taking expectation over the stream/history (iterated expectation) gives unbiasedness; typed and stratified cases are identical per stratum.

4.2 Stratified Sampling and Adaptive Allocation

We partition wedges by center-degree quantiles and recency gap bins; each stratum s keeps its own bottom-k reservoir with inclusion rate $\pi_{s,t} = B_s(t)/A_{s,t}$. Budgets $B_s(t)$ may be fixed proportions or simple functions of $A_{s,t}$ (e.g., floor + cap). Because inclusion is computed per stratum at close time, HT unbiasedness holds regardless of (history-measurable) allocation.

4.3 Variance and Confidence Intervals

Under (approx.) Bernoulli inclusion with rate π, RelVar(HT) $\sim (1-\pi)/\pi$, so allocating budget toward high-variance strata reduces error. **Moving block bootstrap.** We aggregate HT contributions into edge-count blocks of length L and resample blocks ($B^{\star}$=400) to form percentile CIs (per-family and total). We match L to the report cadence (e.g., $L \in \{2000, 5000\}$); standard moving/stationary variants apply.
We select block length L to match the report cadence; automatic selectors for dependent bootstrap are also available [17].

4.4 Diagnostics, Invariants, and Failure Modes

We monitor inclusion rates π, reservoir occupancy, and HT-weight spikes to catch tail volatility; close-once and active-count invariants are enforced. In practice, we find that a slightly larger B or stratified allocation stabilizes rare-family variance.

5 Experiments

5.1 Datasets and Setup

We evaluate on three public temporal graphs: CollegeMsg (student messages), email-Eu (institutional email), and BitcoinOTC (Bitcoin OTC trust). For each, we report number of edges, nodes, and time span. Typed experiments use per-edge `etype` when available.

Sources. CollegeMsg, email-Eu, and BitcoinOTC are available via SNAP [11].

Windows, budgets, and reporting. We study $\Delta \in \{600\,\mathrm{s}, 1800\,\mathrm{s}, 3600\,\mathrm{s}, 7200\,\mathrm{s}\}$ (for CollegeMsg) and Δ=3600 s (for email-Eu/BitcoinOTC), with memory budgets $B \in \{750, 1000, 1500, 2000\}$. Reports are emitted every K=5000 edges.

5.2 Baselines and Metrics

Samplers and uncertainty. We compare *Uniform* vs. *Stratified* (degree-quantiles at 0.5, 0.9; gap bins at 0.33, 0.66 of Δ). We form percentile CIs using a moving block bootstrap with $B^{\star}$=400 replicates and block length $L \in \{2000, 5000\}$ edges to match the reporting cadence.

Ground truth and metrics. Ground truth uses exact *closure events* under close-once semantics on the given prefix (30k for CollegeMsg). We report median relative error (medRel), NRMSE, and empirical 95% coverage of CIs.

5.3 Main Results

We first report accuracy at 30k edges for $\Delta = 3600\,\text{s}$ across budgets and seeds, then sweep Δ at fixed B, and finally present typed results.

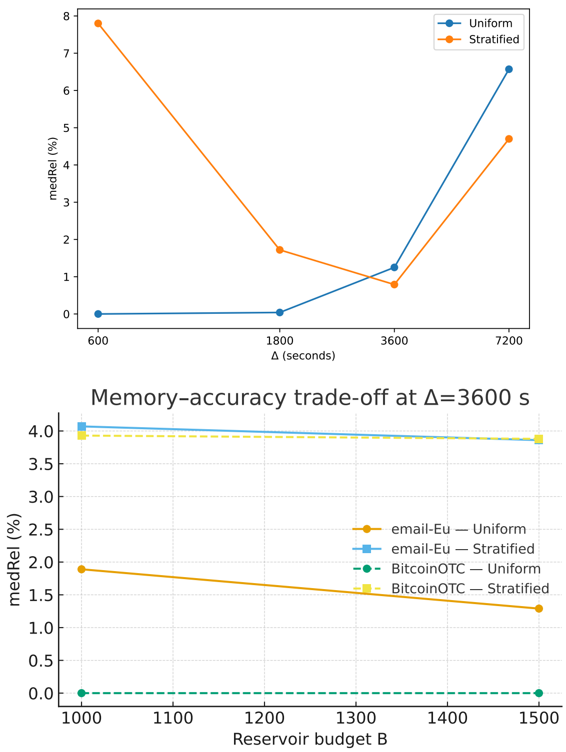

Fig. 1. medRel vs. Δ on CollegeMsg (30k, B=1500); and memory–accuracy trade-off at Δ=3600 s on email-Eu/BitcoinOTC. Lower is better. Empirical 95% coverage is often conservative.

5.4 Ablation Studies

Window-size sweep (CollegeMsg, 30k prefix, $B = 1500$) We examine two questions: (i) sensitivity to the sliding window Δ on CollegeMsg, and (ii) cross-dataset behavior at a fixed window. Table 2 summarizes the window sweep; Table 3 reports additional datasets.

Table 2. CollegeMsg (30k, $B = 1500$): Accuracy vs. Δ (3 seeds). Lower is better for medRel/NRMSE; higher is better for coverage.

Δ (s)	Sampler	medRel (%)	NRMSE (%)
600	Uniform	≤ 0.05	≤ 0.05
600	Stratified	7.80	7.79
1800	Uniform	0.04	0.05
1800	Stratified	1.72	2.26
3600	Uniform	1.25	1.30
3600	Stratified	0.79	0.71
7200	Uniform	6.57	6.93
7200	Stratified	4.70	4.16

Observations. Uniform excels for shorter windows (600–1800 s), while simple degree×gap stratification starts to help at larger windows (3600–7200 s) by stabilizing inclusion probabilities. Very small errors are reported as $\leq 0.05\,\%$. Empirical 95% CIs are near-nominal; $\approx 100\,\%$ indicates conservative coverage due to block calibration.

Table 3. Accuracy on additional datasets at $\Delta = 3600$ s (means over 3 seeds). Coverage is empirical 95% moving-block bootstrap.

Dataset	Sampler	B	medRel (%)	NRMSE (%)	Coverage (%)
email-Eu	Uniform	1000	1.89	1.89	$\approx 100^{\dagger}$
	Uniform	1500	1.29	1.29	$\approx 100^{\dagger}$
	Stratified	1000	4.07	4.07	$\approx 100^{\dagger}$
	Stratified	1500	3.86	3.86	$\approx 100^{\dagger}$
BitcoinOTC	Uniform	1000	≤ 0.05	≤ 0.05	$\approx 100^{\dagger}$
	Uniform	1500	≤ 0.05	≤ 0.05	$\approx 100^{\dagger}$
	Stratified	1000	3.93	3.93	$\approx 100^{\dagger}$
	Stratified	1500	3.88	3.88	$\approx 100^{\dagger}$

†Conservative: block-length and cadence choices can slightly inflate nominal coverage.

Takeaways. Across datasets at Δ=3600 s, errors stay in the low single digits; Uniform is a strong default at modest B, with Stratified providing small benefits when wedge populations are more volatile.

Typed results (CollegeMsg typed, 30k, Δ=600 s) With B=1000 (3 seeds), Uniform attains medRel 1.95% (NRMSE 2.14%) and Stratified 0.78% (NRMSE 0.95%), both with empirical 95% coverage.

5.5 Throughput

On CollegeMsg (30k edges, Δ=3600 s, B=1500, Uniform) we process ∼648 edges/s with 214 MiB peak RSS (measured via `/usr/bin/time -v`). Performance is dominated by Python hash-table ops; a native C++/Rust port should yield order-of-magnitude gains at the same B.

5.6 Semantic Stream Analytics

This section is an *illustrative case study* (not a full ML benchmark) on whether motif features produced by TempoTriads add incremental predictive value. We export per-pair and per-node features within window Δ (closure counts, rates, and family fractions) using the `features` CLI (Subsection 5.7). As a simple downstream task we consider *reciprocity prediction* on CollegeMsg and email-Eu.

Protocol. We build time-aware train/test splits, train a logistic regression baseline on degree/recency features, and evaluate ROC-AUC and Average Precision (AP). We then augment the baseline with *+motif* features (pair/node closure totals, family fractions, and rates). To keep the focus on the estimator rather than classifier tuning, we use default regularization and no feature engineering beyond standardization.

Table 4. Illustrative case study: reciprocity prediction at Δ=600 s. We report ROC-AUC and AP for a degree/recency baseline and with *+motif* features (pair/node motif totals, family fractions, and rates).

Dataset	AUC (base)	AUC (+motif)	AP (base)	AP (+motif)
CollegeMsg	0.698	**0.706**	0.841	**0.846**
email-Eu	0.707	**0.760**	0.916	**0.932**

Findings. We observe small but consistent gains on both datasets (Table 4): +0.008 AUC on CollegeMsg and +0.053 AUC on email-Eu. Because the label distribution is skewed (positives ≈0.63 for CollegeMsg and ≈0.79 for email-Eu), AP is high; we therefore treat ROC-AUC as the base-rate–robust metric. These results indicate that typed triadic closures provide semantic signal beyond degree/recency features. Reaching state-of-the-art task performance is out of scope and left to future work; our goal here is plausibility evidence and an end-to-end reproducible example.

5.7 Reproducibility

Code, configs, and scripts to reproduce *all* table results are available at: `https://github.com/SV25-22/TempoTriads`. The README provides one-command entry points for: (i) exact baselines, (ii) accuracy sweeps (Fig. 1, Table 2), (iii) additional datasets (Table 3), (iv) typed results (Sec. 5.4), (v) semantic analytics (Table 4), and (vi) throughput/profiling. Unless stated otherwise we use Python 3.10 on Linux and seeds $\{0, 1, 2\}$ with `PYTHONHASHSEED=0`.

6 Discussion

6.1 Limitations

We note five limitations. (i) Our default degree×gap strata can be suboptimal at very small Δ under heavy-tailed activity; rare families may show higher variance even though the estimator remains unbiased. (ii) We estimate *closure-event* totals (rates) within Δ, not counts of unique triplets; practitioners should interpret indicators accordingly. (iii) Bootstrap calibration depends on block length roughly matching the reporting cadence; we therefore observe conservative coverage when inclusion rates are high or snapshots are few. (iv) In typed runs, missing/noisy edge types degrade accuracy; we treat types as observed rather than inferred. (v) Throughput reflects a single-threaded Python prototype dominated by hash maps; practicality is demonstrated, but system limits are not.

6.2 Ethics and Privacy Considerations

We use public datasets under their licenses, do not attempt de-anonymization, and report only aggregate statistics. When applying TempoTriads to proprietary data, operators should pseudonymize identifiers, enforce strict access controls, and bound retention by policy. Because high-degree actors can disproportionately influence sampling variance and alert thresholds, fairness audits are recommended before using motif rates for moderation or enforcement.

7 Conclusion and Future Work

We introduced TempoTriads, a streaming estimator for temporal triad closures under a sliding window, combining a bottom-k active-wedge reservoir with Horvitz–Thompson weighting and block-bootstrap uncertainty. The method supports directed/undirected and typed settings, offers stratified and adaptive allocation, and provides diagnostics for inclusion stability. On three public graphs, TempoTriads attains low error with conservatively calibrated coverage at modest budgets.

Beyond technical contributions, TempoTriads facilitates new forms of social-science inquiry by making temporal social structure measurable in real-time streaming contexts. This bridges fine-grained interaction data and classical theories of network formation and evolution, enabling aggregate, uncertainty-aware monitoring aligned with governance and policy needs.

Future work includes (i) priority/PPS sampling for rare families, (ii) richer typed heterogeneity and signed/weighted edges, (iii) distributed and batched streaming deployments and (iv) a native C++ port with contiguous ring buffers for recent-neighbor queues and a fixed-capacity binary heap for bottom-k, plus thread-parallel ingestion; the algorithmic semantics are unchanged, but we expect 5×–15× higher throughput from reduced hash-map overhead and improved cache locality.

Broader societal impact. By turning triadic closure mechanisms into aggregate, uncertainty-aware telemetry, TempoTriads supports descriptive monitoring of conversational health, coordination, and reciprocity without user-level storage. This enables governance and policy experimentation (e.g., rate-limit changes, onboarding workflows) while keeping the unit of analysis at the motif family level, thereby reducing individual-level exposure.

Acknowledgments. We gratefully acknowledge computational support from Xinming Wang at the Institute of Automation, Chinese Academy of Sciences (CASIA). We also thank the Big Data and Networking (DataNET) Group of the SONIC Laboratory for their generous academic guidance, insightful discussions, and continuous support throughout this study.

Disclosure of Interests. The author has no competing interests to declare that are relevant to the content of this article.

References

1. Cai, X., Ke, X., Wang, K., Chen, L., Zhang, T., Liu, Q., Gao, Y.: Efficient temporal butterfly counting and enumeration on temporal bipartite graphs. Proceedings of the VLDB Endowment (PVLDB) **17**(4), 657–670 (2023). `https://doi.org/10.14778/3636218.3636223`
2. Chen, J., Ying, R.: TempME: Towards the explainability of temporal graph neural networks via motif discovery. In: Advances in Neural Information Processing Systems 36 (NeurIPS 2023) (2023), `https://arxiv.org/abs/2310.19324`, neurIPS 2023
3. Cohen, E.: Bottom-k sketches and heavy hitters. ACM SIGMETRICS Performance Evaluation Review **43**(1), 70–72 (2015). `https://doi.org/10.1145/2796314.2796325`
4. Duffield, N., Lund, C., Thorup, M.: Priority sampling for estimation of arbitrary subset sums. Journal of the ACM **54**(6), 32:1–32:38 (2007). `https://doi.org/10.1145/1284320.1284323`
5. Gao, Z., Cheng, C., Yu, Y., Cao, L., Huang, C., Dong, J.: Scalable motif counting for large-scale temporal graphs. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE). IEEE (2022). `https://doi.org/10.1109/ICDE53745.2022.00244`
6. Gou, X., Zou, L.: Sliding window-based approximate triangle counting over streaming graphs with duplicate edges. In: Proceedings of the 2021 ACM SIGMOD International Conference on Management of Data (SIGMOD '21). pp. 645–657. ACM (2021). `https://doi.org/10.1145/3448016.3452800`
7. Holme, P., Saramäki, J.: Temporal networks. Physics Reports **519**(3), 97–125 (2012). `https://doi.org/10.1016/j.physrep.2012.03.001`
8. Horvitz, D.G., Thompson, D.J.: A generalization of sampling without replacement. Journal of the American Statistical Association **47**(260), 663–685 (1952)
9. Jha, M., Seshadhri, C., Pinar, A.: A space-efficient streaming algorithm for triangle counting using the birthday paradox. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). pp. 589–597. ACM (2013). `https://doi.org/10.1145/2487575.2487675`
10. Künsch, H.R.: The jackknife and the bootstrap for general stationary observations. Annals of Statistics **17**(3), 1217–1241 (1989)

11. Leskovec, J., Krevl, A.: Snap datasets: Stanford large network dataset collection. https://snap.stanford.edu/data/ (2014)
12. Li, J., Qi, J., Huang, Y., Cao, L., Yu, Y., Dong, J.: Motto: Scalable motif counting with time-aware topology constraint for large-scale temporal graphs. In: Proceedings of CIKM. pp. 1195–1204 (2024). https://doi.org/10.1145/3627673.3679694
13. Liu, P., Benson, A.R., Charikar, M.: Sampling methods for counting temporal motifs. In: Proceedings of the 12th ACM International Conference on Web Search and Data Mining (WSDM '19). pp. 294–302. ACM (2019). https://doi.org/10.1145/3289600.3290988
14. Pan, Y., Bhalerao, O., Seshadhri, C., Talati, N.: Fast streaming algorithm for counting temporal motifs: A black-box approach. arXiv preprint arXiv:2409.08975 (2024)
15. Paranjape, A., Benson, A.R., Leskovec, J.: Motifs in temporal networks. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM). pp. 601–610. ACM (2017). https://doi.org/10.1145/3018661.3018731
16. Politis, D.N., Romano, J.P.: The stationary bootstrap. Journal of the American Statistical Association **89**(428), 1303–1313 (1994)
17. Politis, D.N., White, H.: Automatic block-length selection for the dependent bootstrap. Econometric Reviews **23**(1), 53–70 (2004). https://doi.org/10.1081/ETC-120028836
18. Stefani, L.D., Epasto, A., Riondato, M., Upfal, E.: TRIÈST: Counting local and global triangles in fully-dynamic streams with fixed memory size. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). pp. 825–834 (2016). https://doi.org/10.1145/2939672.2939771
19. Stefani, L.D., Epasto, A., Riondato, M., Upfal, E.: TRIÈST: Counting triangles in fully-dynamic streams with fixed memory size. ACM Transactions on Knowledge Discovery from Data (TKDD) **11**(4), 43:1–43:50 (2017). https://doi.org/10.1145/3059194
20. Vitter, J.S.: Random sampling with a reservoir. ACM Transactions on Mathematical Software **11**(1), 37–57 (1985). https://doi.org/10.1145/3147.3165