CONTINUOUS-TIME DYNAMIC GRAPH CONDENSATION FOR TEMPORAL EDGE CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

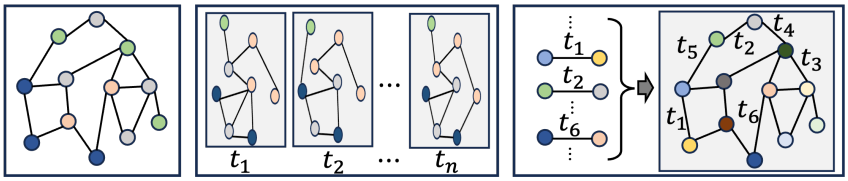
Graph condensation has emerged as a promising technique for distilling large-scale graphs into compact yet informative ones. However, existing approaches primarily focus on static graphs and largely ignore continuous-time dynamic graphs (CTDGs), where graph evolution is represented as asynchronous event streams. Condensing CTDGs is particularly challenging because both structural evolution and temporal dynamics must be preserved under stringent compression constraints. To address this challenge, we propose Continuous-Time Dynamic Graph Condensation (CTDGC), the first graph condensation framework for event-stream continuous-time dynamic graphs and temporal edge classification. CTDGC adopts a learnable selection-based condensation paradigm that assigns each event an importance score and identifies informative events for condensation. To preserve knowledge from the original event stream, CTDGC first trains a lightweight relay model and then optimizes the condensed graph through feature-level and task-level preservation objectives, retaining both temporal representations and task-relevant semantic information. Furthermore, CTDGC employs class-wise top-k event selection to construct condensed graphs. Since the top-k operation is non-differentiable, a Straight-Through Estimator (STE) is introduced to enable end-to-end optimization. This design naturally satisfies arbitrary compression budgets without requiring additional sparsity-inducing constraints. In extensive experiments on multiple benchmarks, CTDGC consistently outperforms compared baselines under various compression ratios and generalizes well across diverse temporal graph learning architectures. Moreover, on several datasets, models trained on CTDGC-condensed graphs achieve performance competitive with those trained on the original full event streams while using only a small fraction of the training events. Our code will be shared publicly upon paper acceptance.

1 INTRODUCTION

The rapid growth of real-world graph datasets has posed significant scalability challenges for graph learning, particularly in scenarios that require repeated model training, such as hyperparameter optimization (Dedic & Belohlávek, 2026), neural architecture search (Xue et al., 2026), and continual graph learning (Yuan et al., 2026). To alleviate these computational and storage burdens, graph condensation has emerged as a promising data-centric paradigm (Gao et al., 2025b), aiming to construct a compact yet representative graph that preserves the essential information of a large original graph. Ideally, models trained on the condensed graph should achieve performance comparable to those trained on the full graph, while substantially accelerating model development. Existing graph condensation methods (Fang et al., 2024; Jin et al., 2022; Zheng et al., 2023; Zhang et al., 2024c; Liu et al., 2024a; Gao et al., 2025a; Xu et al., 2023) optimize the condensed graph through different information-preserving principles (see Sec. 2 for more details).

Despite their success, previous methods are primarily developed for static graphs, where graph structures remain unchanged over time (Figure 1(a)). However, many real-world systems evolve continuously over time and are more naturally represented as continuous-time dynamic graphs (CTDGs), where events or interactions are recorded as asynchronous event streams. Condensing such graphs is considerably more challenging. First, effective condensation must preserve both structural evolution and temporal dynamics, capturing not only *who* interacted with *whom*, but also *when* and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) static graph (b) graphs in DyGC (c) graph in our study

Figure 1: Different graph patterns in previous works and our study (time stamps are on edges).

in *what order* the interactions occurred. Second, CTDGs often contain highly irregular and dense event streams, resulting in substantial spatiotemporal redundancy. Furthermore, many downstream tasks on CTDGs, such as link prediction and edge classification, are inherently edge-centric. Under extreme compression ratios, an effective condensation method should preserve not only the overall graph structure but also the distributional and temporal characteristics associated with different categories of interactions. Recently, DyGC (Chen et al., 2026) extended graph condensation to dynamic settings by constructing a condensed counterpart for each graph snapshot, resulting in a compact sequence of temporal graph snapshots for temporal node classification (Figure 1(b)). While effective, DyGC relies on graph discretization, which inevitably sacrifices fine-grained temporal information and limits its ability to model continuously evolving event streams. Moreover, its design is specifically tailored to node-level tasks and cannot be directly applied to continuous-time edge classification, where preserving interaction semantics and temporal dependencies is particularly crucial.

To address these limitations, we propose **Continuous-Time Dynamic Graph Condensation (CTDGC)**, the first graph condensation framework designed for event-stream CTDGs and temporal edge classification tasks (Figure 1(c)). Inspired by distribution matching-based graph condensation on static graphs (Xiao et al., 2025; Singh et al., 2026), CTDGC preserves the knowledge of the original event stream through both feature- and task-level preservation. Unlike previous graph condensation methods that directly optimize synthetic graph structures and attributes through carefully designed objectives, CTDGC adopts a learnable selection-based condensation paradigm. Specifically, we first train a lightweight relay model on the original event stream to capture the underlying temporal interaction patterns. After the relay model is trained, each event is assigned a learnable importance score that reflects its contribution to the predictive knowledge of the original graph.

Under a given compression ratio, an ideal condensed graph should contain only a sparse subset of informative events while preserving the desired number of events for each class. Directly enforcing such constraints typically requires multiple auxiliary objectives and carefully designed regularization terms. To avoid this complexity, CTDGC constructs the condensed graph by directly selecting the top-k highest-scored events within each class, where k is determined by the compression ratio. Since the top-k operation is non-differentiable, we employ a Straight-Through Estimator (STE) (Yin et al., 2019) to enable gradient propagation through the discrete selection process. This allows event importance scores to be optimized end-to-end while maintaining hard event selection in the forward pass.

Given the original event stream and the current condensed event stream, both graphs are processed by the frozen relay model. The selected events are then optimized through feature-level and task-level preservation objectives, which encourage the condensed graph to retain temporal representations and task-relevant semantic information from the original event stream. After optimization, events with the highest learned importance scores are retained to form a compact yet informative condensed graph. The resulting graph can subsequently be used to train downstream temporal edge classification models with substantially reduced storage and computational costs while maintaining competitive predictive performance.

The main contributions of this work are summarized as follows:

(1) To the best of our knowledge, this work presents the first graph condensation framework for continuous-time event-stream graphs (CTDGC), which extends graph condensation from static graphs and snapshot-based dynamic graphs to continuous-time dynamic graph data and temporal edge classification tasks.

(2) In the proposed CTDGC framework, we introduce a learnable selection-based condensation paradigm that constructs condensed graphs through differentiable top-k event selection. By leveraging a straight-through estimator, CTDGC enables end-to-end optimization of discrete event selection without requiring complex sparsity or budget-constrained objectives.

(3) We develop a dual-level knowledge preservation strategy that aligns both latent event representations and prediction outputs between the original and condensed event streams, allowing the condensed graph to effectively retain temporal interaction patterns and category-specific semantics.

(4) Extensive experiments on multiple benchmark datasets demonstrate the effectiveness of CTDGC under various compression ratios. CTDGC consistently outperforms competitive adapted baselines while maintaining performance comparable to that obtained using the original full event streams.

2 Related Works

Continuous-Time Dynamic Graph Learning has attracted considerable attention recently due to its ability to model evolving relational systems and asynchronous interaction events (Zhang et al., 2026; Feng et al., 2026). Existing methods primarily focus on designing expressive temporal architectures to capture structural and temporal dependencies. Early studies relied on recurrent architectures, such as JODIE (Kumar et al., 2019), which updates node representations after each interaction event. Later works introduced attention-based mechanisms, including TGAT (Xu et al., 2020) and DyGFormer (Yu et al., 2023), to better capture long-range temporal dependencies and continuously evolving interaction patterns. More recently, scalable temporal graph learning frameworks such as EasyDGL (Chen et al., 2024) and SEAN (Zhang et al., 2024b) have been proposed to improve the efficiency and effectiveness of continuous-time graph representation learning. Recent studies have further explored richer graph modalities beyond temporal interactions. For example, MoMent (Xu et al., 2026) explicitly models and aligns temporal, textual, and structural modalities in dynamic text-attributed graphs through a multimodal representation learning framework, demonstrating the benefits of integrating heterogeneous information sources for temporal graph learning.

Although these methods have significantly advanced continuous-time dynamic graph learning, they are mainly model-centric and often require repeated training on large event streams. In contrast, our work explores a complementary data-centric direction by reducing the size of continuous-time dynamic graphs through graph condensation.

Graph Condensation aims to construct a compact graph that preserves the essential knowledge of a large original graph while significantly reducing training cost. Existing methods can generally be categorized according to their optimization objectives. Gradient matching methods (Feng et al., 2023; Fang et al., 2024) optimize synthetic graphs by aligning model gradients computed on condensed and original data. Trajectory matching approaches (Zhang et al., 2024c; Liu et al., 2024b) extend this idea by matching multi-step optimization trajectories, thereby preserving richer training dynamics. To avoid expensive bi-level optimization, distribution matching methods (Xiao et al., 2025; Singh et al., 2026) directly align feature or representation distributions between condensed and original graphs in a latent space. In addition, kernel ridge regression-based methods (Gao et al., 2024b; Wang & Li, 2026) replace graph neural networks with graph neural tangent kernels, enabling more efficient optimization through closed-form solutions.

As graph condensation continues to evolve, recent studies have expanded beyond homogeneous static graphs toward more complex graph structures. For example, heterogeneous graph condensation methods (Gao et al., 2024a; Liang et al., 2025) preserve multi-relational topological information through dedicated relational modeling mechanisms. More recently, DyGC (Chen et al., 2026) extended graph condensation to dynamic graphs by aligning spatiotemporal representations across graph snapshots and generating compact dynamic graph sequences.

Despite these advances, several important challenges remain unresolved. First, most existing graph condensation methods are designed for node-level tasks, while edge-level tasks, such as temporal edge classification, have received little attention. Second, existing dynamic graph condensation frameworks, represented by DyGC (Chen et al., 2026), operate on discrete-time graph snapshots. As a result, they cannot fully preserve the fine-grained temporal information contained in continuous event streams, making them difficult to apply directly to event-stream graph condensation scenarios. These limitations motivate us to investigate graph condensation for continuous-time dynamic graph.

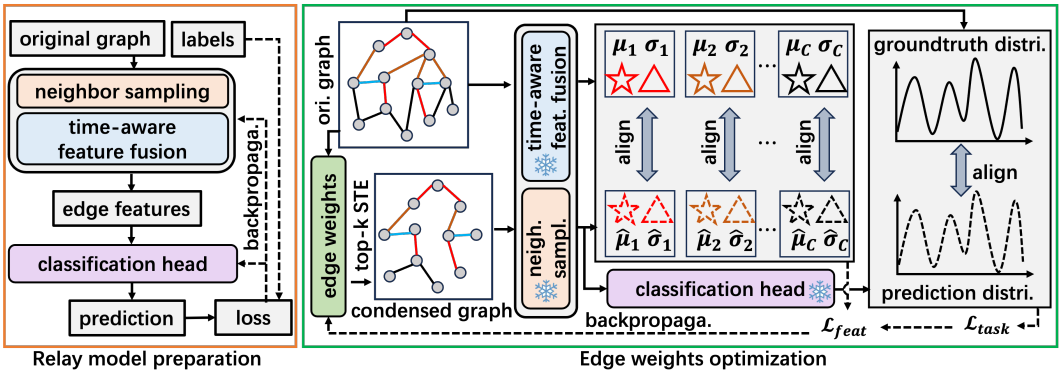


Figure 2: Overview of the proposed framework.

3 METHOD

3.1 PROBLEM FORMULATION

Continuous-Time Dynamic Graph. A continuous-time dynamic graph (CTDG) is commonly represented as a stream of timestamped interaction events (Feng et al., 2026). Formally, a CTDG over a time interval $[0, T]$ is defined as $\mathcal{G} = \{\epsilon_1, \epsilon_2, \ldots, \epsilon_M\}$, where M denotes the total number of interaction events. Each event $\epsilon_k = (u_k, v_k, t_k, \mathbf{e}_k)$ represents an interaction occurring at timestamp $t_k \in [0, T]$ between source node $u_k \in \mathcal{V}$ and target node $v_k \in \mathcal{V}$. Here, $\mathcal{V}$ denotes the node set, and $\mathbf{e}_k \in \mathbb{R}^{d_E}$ is the corresponding event (edge) attribute vector, where d_E is the dimensionality of event attributes.

Temporal Edge Classification. Given a node $u \in \mathcal{V}$, its historical interaction sequence before time t is defined as $\mathcal{H}_u(t) = \{\epsilon_k = (u_k, v_k, t_k, \mathbf{e}_k) \in \mathcal{G} \mid (u_k = u \vee v_k = u) \wedge t_k < t\}$. A temporal representation learning model f_{rep} maps node u into a time-dependent embedding: $\mathbf{z}_u(t) = f_{\text{rep}}\big(u, t, \mathcal{H}_u(t); \Theta_{\text{rep}}\big) \in \mathbb{R}^d$, where Θ_{rep} denotes the model parameters. For an interaction event $\epsilon = (u, v, t, \mathbf{e}) \in \mathcal{G}$, the temporal edge classifier predicts its category based on the temporal embeddings of the interacting nodes and the associated edge feature: $\hat{y} = f_{\text{cls}}\big(\mathbf{z}_u(t), \mathbf{z}_v(t), \mathbf{e}; \Theta_{\text{cls}}\big)$, where Θ_{cls} denotes the classifier parameters and $\hat{y} \in \mathcal{Y} = \{1, 2, \ldots, C\}$ is the predicted edge label among C categories.

Graph Condensation for CTDG. Given an original CTDG $\mathcal{G}_{\text{ori}} = \{(\epsilon_k, y_k)\}_{k=1}^{M_{\text{ori}}}$, where $y_k \in \mathcal{Y}$ denotes the label of event ϵ_k, the goal of graph condensation for CTDG is to construct a compact event subset $\mathcal{G}_{\text{syn}} = \{(\epsilon'_r, y'_r)\}_{r=1}^{M_{\text{syn}}}$, where $M_{\text{syn}} \ll M_{\text{ori}}$, while preserving the predictive knowledge contained in the original graph. Specifically, a temporal edge classification model trained on the condensed graph should achieve performance comparable to a model trained on the full graph. Formally, the condensation objective can be expressed as $\mathcal{G}^*_{\text{syn}} = \arg\min_{\mathcal{G}_{\text{syn}}} \mathcal{L}_{\text{test}}\Big(\Theta^*; \mathcal{G}_{\text{test}}\Big)$, where $\Theta^* = \arg\min_{\Theta} \mathcal{L}_{\text{train}}\Big(\Theta; \mathcal{G}_{\text{syn}}\Big)$, and $\Theta = \{\Theta_{\text{rep}}, \Theta_{\text{cls}}\}$ collectively denotes the parameters of the temporal representation learner and the edge classifier. $\mathcal{L}_{\text{train}}$ and $\mathcal{L}_{\text{test}}$ denote the task-specific training and testing objectives, respectively (e.g., multi-class cross-entropy loss for edge classification).

3.2 FRAMEWORK OVERVIEW

Figure 2 illustrates the overall architecture of the proposed Continuous-Time Dynamic Graph Condensation framework (CTDGC). Unlike conventional graph condensation methods that directly optimize synthetic graph structures and attributes, CTDGC adopts a learnable selection-based condensation paradigm tailored for continuous-time dynamic graphs. Given an original continuous-time dynamic graph $\mathcal{G}_{\text{ori}}$, CTDGC first trains a lightweight relay model consisting of a temporal rep-

resentation learner and an edge classification head. After training, all relay model parameters are frozen and serve as a fixed source of knowledge during the condensation process. Each event in the original event stream is then associated with a learnable importance score. Based on these scores, CTDGC iteratively constructs a condensed graph under a specified compression ratio through class-wise event selection. The original graph and the current condensed graph are subsequently processed by the frozen relay model to produce event representations and prediction outputs. To preserve the knowledge contained in the original event stream, CTDGC employs two complementary objectives. A feature-level preservation loss ($\mathcal{L}_{\text{feat}}$) encourages the condensed graph to maintain the temporal and structural characteristics captured by the relay model, while a task-level preservation loss ($\mathcal{L}_{\text{task}}$) preserves task-relevant semantic information for temporal edge classification. By jointly optimizing these objectives, CTDGC identifies informative events and removes redundant interactions, resulting in a compact yet effective condensed graph for downstream temporal graph learning.

3.3 Relay Model Preparation

The relay model serves as a fixed knowledge extractor throughout the condensation process, providing event representations and task supervision for optimizing event importance scores. To reduce the influence of model-specific inductive biases and improve optimization stability, CTDGC adopts a lightweight temporal edge classification architecture as the relay model.

Given an original continuous-time dynamic graph $\mathcal{G}_{\text{ori}}$, the relay model consists of three components: a neighbor sampler, a time-aware feature fusion module, and an edge classification head. For an event $\epsilon_k = (u_k, v_k, t_k, \mathbf{e}_k)$, the neighbor sampler first constructs its local temporal context. Specifically, only historical events that occurred before the current timestamp t_k are considered. For each interacting node, the sampler retrieves a fixed number of the most recent historical interactions to form a local temporal neighborhood. The resulting neighborhood provides the contextual information used by the subsequent time-aware feature fusion module. Let $\mathbf{h}_{u_k} \in \mathbb{R}^d$, $\mathbf{h}_{v_k} \in \mathbb{R}^d$, and $\mathbf{h}_{e_k} \in \mathbb{R}^d$ denote the feature representations of the source node, target node, and edge associated with event ϵ_k, respectively. The event feature is constructed as $\mathbf{x}_k = [\mathbf{h}_{u_k} \parallel \mathbf{h}_{e_k} \parallel \mathbf{h}_{v_k}]$. To model temporal dynamics without introducing additional trainable parameters, the time-aware feature fusion module adopts an exponential time-decay aggregation strategy. For a node u queried at time t, its temporal representation is computed as

$$\mathbf{z}_u(t) = f_{\text{rep}}\big(u, t, \mathcal{H}_u(t)\big) = \sum_{\epsilon_k \in \mathcal{H}_u(t)} \frac{\exp[-\lambda(t - t_k)]}{\sum_{\epsilon_m \in \mathcal{H}_u(t)} \exp[-\lambda(t - t_m)]} \mathbf{x}_k, \tag{1}$$

where $\lambda > 0$ is a predefined decay coefficient controlling the influence of historical interactions over time. More recent interactions therefore contribute more heavily to the resulting representation. This parameter-free temporal aggregation enables the relay model to remain lightweight while effectively capturing temporal evolution patterns. For an interaction event $\epsilon_k = (u_k, v_k, t_k, \mathbf{e}_k)$, the final event representation is constructed as $\mathbf{h}_k = [\mathbf{z}_{u_k}(t_k) \parallel \mathbf{z}_{v_k}(t_k)]$. The resulting event representation is fed into an edge classification head implemented as a multilayer perceptron (MLP) to predict the event category. The relay model is trained on the original event stream using the standard multi-class cross-entropy objective. After training, all relay model parameters are frozen and remain fixed throughout the condensation stage. The frozen relay model subsequently provides event representations for feature-level preservation and task supervision for task-level preservation during event-weight optimization.

3.4 Event-Weight Optimization

With the relay model fixed, the goal of CTDGC is to identify the most informative events for graph condensation. To this end, we associate each event $\epsilon_k \in \mathcal{G}_{\text{ori}}$ with a learnable importance score $w_k \in \mathbb{R}$. The magnitude of w_k reflects the relative importance of event ϵ_k during condensation, with larger values indicating a higher likelihood of being selected into the condensed graph. Let $\mathbf{w} = [w_1, w_2, \ldots, w_M]^\top$ denote the collection of learnable event importance scores.

Class-wise Top-k Event Selection. Given a target compression ratio $r \in (0, 1)$, the condensed graph should retain only a small subset of informative events while preserving the original class distribution. Instead of introducing additional sparsity-inducing or budget-constrained objectives, CTDGC directly constructs the condensed graph through a class-wise top-k selection strategy. Let

$\mathcal{E}_{ori}^{(c)} = \{\epsilon_k \in \mathcal{G}_{\text{ori}} \mid y_k = c\}$ denote the set of events belonging to class c. For each class, the number of retained events is determined by $K_c = \left\lfloor r|\mathcal{E}_{ori}^{(c)}| \right\rfloor$. Events within each class are ranked according to their importance scores, and the top-K_c events are selected to construct the condensed graph: $\mathcal{G}_{\text{syn}} = \bigcup_{c=1}^{C} \text{TopK}\left(\mathcal{E}_{ori}^{(c)}, K_c\right)$. This strategy naturally satisfies the prescribed compression ratio while preserving the class distribution of the original event stream. Since the top-k operator is discrete and non-differentiable, gradients cannot be directly propagated through the selection process. To enable end-to-end optimization, we employ a Straight-Through Estimator (STE) (Yin et al., 2019). Let $\mathbf{m}$ denote the binary top-k selection mask. We construct a differentiable mask as $\tilde{\mathbf{m}} = (\mathbf{m} - \mathbf{w})_{\text{detach}} + \mathbf{w}$. During the forward pass, $\tilde{\mathbf{m}}$ is numerically identical to the hard selection mask $\mathbf{m}$, ensuring that only selected events participate in subsequent computations. During the backward pass, gradients bypass the non-differentiable top-k operation and are directly propagated through the importance scores $\mathbf{w}$. Consequently, the event importance scores can be optimized end-to-end while maintaining discrete event selection throughout training.

Feature-level Preservation. Given the original graph $\mathcal{G}_{\text{ori}}$ and the current condensed graph $\mathcal{G}_{\text{syn}}$, both graphs are processed by the frozen relay model to obtain event representations. To preserve the structural and temporal knowledge captured by the relay model, CTDGC enforces representation consistency between selected events under the original and condensed graph contexts. Specifically, for each class c, let $\mathcal{G}_{\text{syn}}^{(c)} = \{\epsilon_k \in \mathcal{G}_{\text{syn}} \mid y_k = c\}$ denote the set of selected events belonging to class c. For each selected event, let $\mathbf{h}_k$ and $\hat{\mathbf{h}}_k$ denote its relay-model representations obtained under the original and condensed graph contexts, respectively. To preserve the semantic information encoded by the relay model while respecting the inherent class distribution of the original graph, we introduce a class-proportion weight $\rho_c = |\mathcal{G}_{\text{ori}}^{(c)}|/|\mathcal{G}_{\text{ori}}|$, which prevents minority classes from being overwhelmed during feature alignment. Moreover, feature distributions in continuous-time dynamic graphs may exhibit substantial variance during early optimization. To reduce the influence of outlier representations and improve training stability, we align the first-order and second-order statistics using the Huber loss (Huber, 1964):

$$\mathcal{L}_{\text{feat}} = \sum_{c=1}^{C} \rho_c \left(\text{Huber}\left(\boldsymbol{\mu}_c - \hat{\boldsymbol{\mu}}_c\right) + \text{Huber}\left(\boldsymbol{\sigma}_c - \hat{\boldsymbol{\sigma}}_c\right) \right), \tag{2}$$

where $\boldsymbol{\mu}_c = \text{Mean}\left(\{\mathbf{h}_k \mid \epsilon_k \in \mathcal{G}_{\text{syn}}^{(c)}\}\right)$, $\hat{\boldsymbol{\mu}}_c = \text{Mean}\left(\{\hat{\mathbf{h}}_k \mid \epsilon_k \in \mathcal{G}_{\text{syn}}^{(c)}\}\right)$, and $\boldsymbol{\sigma}_c$ and $\hat{\boldsymbol{\sigma}}_c$ denote the corresponding standard deviations computed under the original and condensed graph contexts, respectively. By aligning class-wise feature statistics of the same selected events across two graph contexts, CTDGC encourages the condensed graph to preserve the temporal and structural information originally captured by the full event stream.

Task-level Preservation. Feature preservation alone does not guarantee that the condensed graph retains task-discriminative information. Therefore, CTDGC additionally introduces a task-level preservation loss to preserve the predictive capability of the condensed graph. Specifically, the relay model predicts labels for events contained in the condensed graph, and the predictions are directly supervised using the corresponding ground-truth labels:

$$\mathcal{L}_{\text{task}} = -\sum_{\epsilon_k \in \mathcal{G}_{\text{syn}}} \sum_{c=1}^{C} y_{k,c} \log \hat{y}_{k,c}, \tag{3}$$

where $y_{k,c}$ denotes the c-th element of the one-hot encoding of the ground-truth label of event ϵ_k, and $\hat{y}_{k,c}$ denotes the predicted probability of class c. This objective encourages the condensation process to preserve events that are most informative for temporal edge classification, thereby retaining task-specific semantic knowledge in the condensed graph.

Overall Objective. The final optimization objective combines feature-level preservation and task-level preservation: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{feat}} + \alpha\mathcal{L}_{\text{task}}$, where α controls the trade-off between preserving feature distributions and maintaining task-specific predictive information. During optimization, gradients are propagated through the STE to update the event importance scores, while all relay model parameters remain frozen. After optimization converges, events with the highest learned importance scores are retained to form the final condensed graph, which can subsequently be used to train downstream temporal edge classification models.

4 EXPERIMENTS

We conduct extensive experiments on five public CTDG benchmarks and compare our method against several competitive event selection baselines under different compression ratios.

Datasets. We evaluate CTDGC on five continuous-time dynamic graph datasets from the Dynamic Text-attributed Graph Benchmark (DTGB) (Zhang et al., 2024a), including Enron, ICEWS1819, GDELT, Googlemap CT, and Amazon Movies. These datasets cover diverse application domains, including e-mail communication, temporal knowledge graphs, and e-commerce systems. All datasets are formulated as temporal edge classification tasks, where each event is associated with a timestamp and a categorical edge label. Table 1 summarizes the statistics of the datasets used in our experiments. Following the official DTGB protocol, all datasets are chronologically split into training, validation, and test sets with a ratio of 70%/15%/15%. Graph condensation is performed exclusively on the training partition, while the validation and test partitions remain unchanged throughout all experiments.

Table 1: Statistics of datasets in our experiments.

	Enron	ICEWS.	GDELT	G.CT	A.Mov.
#Nodes	42,711	31,796	6,786	111,168	293,566
#Edges	797,907	1,100,071	1,339,245	1,380,623	3,217,324
#Classes	10	266	237	5	5
Domain	E-mail	K. graph	K. graph	E-comm.	E-comm.

Baselines. To our best knowledge, this is the first study on graph condensation for CTDG and temporal edge classification. Since no existing graph condensation method is directly applicable to this setting, we construct five baselines adapted from representative data selection principles. Random Sampling uniformly selects events from the original event stream according to the target compression ratio while preserving the original class distribution. Segmented Herding and Segmented K-Center extend two classical dataset condensation strategies to continuous-time event streams. Specifically, the training stream is chronologically partitioned into multiple non-overlapping temporal segments, after which Herding selects events whose aggregated feature representations approximate the segment centroid, whereas K-Center greedily chooses representative events to maximize feature-space coverage. Aside from feature-based selection methods, we further adapt two data valuation strategies, *EL2N* and *GraNd*, proposed by Paul et al. (2021). EL2N ranks events according to their early-training prediction errors, whereas GraNd estimates event importance using the gradient norms generated during the early stages during training. Events with the highest scores are retained under the compression budget. These baselines collectively cover random selection, feature-distribution preservation, geometric coverage preservation, prediction-difficulty estimation, and gradient-based data valuation, providing a comprehensive evaluation.

Downstream Evaluation Model. To evaluate the quality of condensed graphs, we train a temporal edge classification model from scratch using only the condensed graph and evaluate it on the test partition of the original graph. The downstream model is implemented using a multi-hop temporal cross-graph attention network. Specifically, it first encodes temporal intervals through a learnable continuous-time encoding module and then performs recursive temporal graph convolutions over sampled historical neighborhoods. To capture higher-order temporal dependencies while mitigating over-smoothing, each convolution layer incorporates an explicit learnable decay mechanism that progressively attenuates information from distant hops. Furthermore, a dual-end cross-attention pooling module is adopted to model interaction semantics between source and destination nodes, by allowing each endpoint to selectively aggregate contextual information from the other's historical neighborhood. The resulting node representations are finally fed into an MLP for temporal edge classification. The same architecture is used for all condensation methods to ensure a fair comparison. Following DTGB (Zhang et al., 2024a), we evaluate all methods using the weighted F1-score. All reported results are averaged over multiple independent runs.

Main Results. Table 2 reports the main experimental results under different compression ratios. Overall, CTDGC achieves the best compressed-graph performance in most evaluation settings, demonstrating the effectiveness of the proposed event importance learning framework. Across five datasets and three compression ratios, CTDGC obtains the highest weighted F1-score in 12 out of 15 settings, consistently outperforming most adapted baselines. The advantage of CTDGC is particularly evident on large and complex temporal datasets. On GDELT, CTDGC consistently achieves the best performance under all compression ratios, improving the strongest baseline from 4.09% to

Table 2: Weighted F1-score (%, mean±std) achieved by different graph condensation methods under various compression ratios. The best compressed results are highlighted in bold.

	Ratio	Random	K-center	Herding	EL2N	GraNd	CTGDC(Ours)	Full
Enron	0.5%	42.65±3.88	37.93±1.95	42.14±0.40	35.26±2.44	36.33±0.64	**45.84±0.94**	
	2.5%	53.35±3.30	51.26±5.54	57.45±1.73	46.02±1.56	54.34±0.39	**59.18±0.73**	60.12±0.17
	5.0%	57.28±0.62	53.99±2.05	57.87±1.14	54.40±0.78	58.39±1.02	**59.25±0.37**	
ICEWS.	0.5%	15.91±0.33	4.84±0.42	18.12±0.72	6.64±0.47	12.12±0.43	**19.07±0.05**	
	2.5%	**23.02±0.19**	17.97±1.22	22.34±0.68	12.58±0.06	19.31±1.21	22.36±0.08	29.37±0.56
	5.0%	23.95±0.61	21.64±0.16	23.47±0.92	12.95±0.31	21.05±0.73	**24.01±0.25**	
GDELT	0.25%	3.31±0.95	2.35±0.86	2.93±0.38	4.09±0.21	2.61±1.47	**5.84±0.36**	
	0.5%	4.86±0.40	2.75±0.79	4.90±0.62	4.41±0.47	3.59±0.53	**6.49±0.27**	9.99±0.05
	2.5%	5.36±0.30	5.73±0.91	5.52±0.20	4.41±0.32	5.83±0.67	**7.46±0.05**	
G. CT	0.25%	52.75±0.00	52.75±0.00	53.07±0.36	51.67±0.77	53.21±0.33	**55.91±0.79**	
	0.5%	54.26±2.60	52.98±0.38	56.38±2.39	52.78±2.73	56.18±0.87	**57.49±0.12**	61.80±0.17
	2.5%	55.58±2.07	54.14±0.67	54.76±1.85	46.11±1.70	**57.57±1.21**	57.33±0.44	
A. Mov.	0.5%	52.23±1.10	51.91±1.04	**56.49±0.25**	51.87±0.28	43.56±4.89	55.16±0.22	
	2.5%	56.15±1.40	51.47±0.10	57.68±0.83	51.56±0.24	44.65±6.93	**57.74±0.11**	58.94±0.95
	5%	56.19±0.67	52.96±0.54	57.91±0.50	51.93±0.22	53.09±1.44	**58.15±0.23**	

Table 3: Cross-model weighted F1-score (%) of different condensation methods.

Dataset	Method	JODIE	DyRep	TGAT	CAWN	TCL	GMixer	MoMent	AVG.
Enron (0.005)	Random	38.82	38.36	36.20	38.73	39.66	39.54	36.75	38.29
	K-center	36.43	34.91	35.69	37.32	37.47	27.19	26.10	33.59
	Herding	37.74	39.87	43.55	37.28	**47.80**	**40.54**	37.59	40.62
	EL2N	**42.19**	**47.16**	36.33	**49.05**	38.47	34.58	35.22	40.43
	GraNd	34.39	33.33	30.25	34.38	37.57	37.41	36.04	34.77
	CTGDC	39.69	41.51	**51.40**	43.43	45.02	37.93	**41.00**	**42.85**
GDELT (0.025)	Random	5.97	6.00	5.31	6.29	4.46	6.34	4.32	5.53
	K-center	**8.14**	**8.34**	4.85	5.42	3.83	6.51	6.34	6.20
	Herding	6.61	6.50	5.94	6.74	4.88	6.29	7.07	6.29
	EL2N	7.67	6.80	4.62	4.19	4.90	5.14	5.04	5.48
	GraNd	7.67	8.01	5.15	6.12	3.70	6.51	6.07	6.18
	CTGDC	7.86	7.67	**7.25**	**7.52**	**6.56**	**7.55**	**7.76**	**7.45**
G. CT (0.0025)	Random	54.14	54.17	52.75	53.26	52.75	52.75	53.07	53.27
	K-center	52.34	50.68	52.75	52.76	53.07	52.91	52.76	52.47
	Herding	53.55	53.18	52.75	54.37	**55.36**	52.75	52.97	53.56
	EL2N	55.30	54.91	53.86	**54.81**	53.86	**53.09**	52.75	54.08
	GraNd	55.46	55.42	54.23	52.85	52.75	52.76	**53.12**	53.80
	CTGDC	**58.11**	**58.26**	**54.74**	53.73	54.08	52.76	52.75	**54.92**
ICEWS1819 (0.005)	Random	12.84	13.79	16.97	16.57	15.47	13.86	8.17	13.95
	K-center	10.47	7.90	6.79	10.69	2.81	2.75	6.80	6.89
	Herding	13.31	13.94	**18.09**	17.79	11.63	15.39	8.30	14.06
	EL2N	10.21	11.93	9.71	11.64	10.29	4.87	6.63	9.33
	GraNd	10.66	13.50	8.11	10.88	5.88	7.41	6.83	9.04
	CTGDC	**18.24**	**18.54**	17.92	**19.66**	**15.94**	**17.59**	**16.52**	**17.77**

5.84% at 0.25% compression and from 5.83% to 7.46% at 2.5% compression. Similar improvements can be observed on Enron, where CTDGC remains highly competitive with the full graph while using only a small fraction of the original events. On ICEWS1819, Googlemap CT, and Amazon Movies, CTDGC consistently ranks among the best-performing methods. Although certain baselines occasionally achieve slightly higher performance under specific compression ratios, CTDGC exhibits the most stable performance across different datasets and storage budgets. These results suggest that jointly preserving feature-level representations and task-specific knowledge enables CTDGC to effectively retain informative temporal interactions while removing substantial redundancy from the original event stream. Another notable observation is that the performance gap between CTDGC and the full graph remains relatively small on several datasets despite aggressive compression. This finding indicates that continuous-time event streams contain considerable redundancy and that carefully selected event subsets can preserve most task-relevant temporal semantics.

Generalization Across Downstream Models. To evaluate the cross-model generalization ability of condensed graphs, we further conduct experiments using seven representative continuous-time dynamic graph learning models, including JODIE (Kumar et al., 2019), DyRep (Trivedi et al., 2019), TGAT (Xu et al., 2020), CAWN (Wang et al., 2021b), TCL (Wang et al., 2021a), GMixer (Cong et al., 2023), and MoMent (Xu et al., 2026). All models are trained from scratch on the condensed graphs and evaluated using the same protocol as in the previous experiments. The results are reported in Table 3. Overall, CTDGC achieves the highest average weighted F1-score on all four datasets, demonstrating that the proposed condensation framework is not tied to a specific downstream model. The condensed graphs generated by CTDGC preserve task-relevant temporal information that can be effectively exploited by diverse temporal graph learning architectures. The superiority of CTDGC is particularly pronounced on GDELT and ICEWS1819. On GDELT, CTDGC achieves the best performance for five out of seven downstream models and obtains the highest average score of 7.45%. On ICEWS1819, CTDGC consistently ranks among the top-performing methods and improves the average weighted F1-score from 14.06% achieved by the strongest baseline to 17.77%. On Enron and Googlemap CT, CTDGC also achieves the highest average performance across all condensation methods while remaining competitive on most individual models. Although competing baselines occasionally outperform CTDGC on specific architectures, CTDGC exhibits substantially stronger overall robustness across different temporal graph learning models. These results demonstrate that the condensed graphs produced by CTDGC generalize well across recurrent-based, attention-based, and attention-driven temporal graph architectures, confirming that CTDGC effectively preserves transferable temporal and semantic information from the original event stream.

Ablation Study. To evaluate different optimization objectives, we perform an ablation study under a 2.5% compression ratio by removing either the task-level preservation loss (w/o $\mathcal{L}_{\text{task}}$) or the feature-level preservation loss (w/o $\mathcal{L}_{\text{feat}}$). The results are reported in Table 4. Overall, the complete CTDGC framework consistently achieves the best performance on all datasets. Removing either objective leads to performance degradation, indicating that the two preservation mechanisms provide complementary information for identifying informative events and constructing high-quality condensed graphs. Among the two variants, removing the task-level preservation loss causes larger performance drops on Enron and GDELT, while removing the feature-level preservation loss has a stronger impact on ICEWS1819, Googlemap CT, and Amazon Movies. These observations suggest that task supervision is important for preserving discriminative semantic information, whereas feature preservation helps retain the structural and temporal characteristics of the original event stream. Combining both objectives allows CTDGC to achieve the most effective graph condensation performance across diverse datasets.

Table 4: Ablation study of feature-level and task-level preservation under 2.5% compression ratio.

Method	Enron	ICEWS18	GDELT	G. CT	A. Movies
w/o $\mathcal{L}_{\text{task}}$	45.40 ± 0.38	22.20 ± 0.10	4.28 ± 0.22	54.86 ± 1.23	52.60 ± 0.28
w/o $\mathcal{L}_{\text{feat}}$	46.22 ± 3.77	20.57 ± 0.24	7.12 ± 0.04	45.78 ± 0.91	48.64 ± 0.30
CTDGC	$\mathbf{61.16 \pm 1.33}$	$\mathbf{22.36 \pm 0.08}$	$\mathbf{7.46 \pm 0.05}$	$\mathbf{57.33 \pm 0.44}$	$\mathbf{57.74 \pm 0.11}$

5 Conclusion

In this paper, we investigate graph condensation for continuous-time dynamic graphs and temporal edge classification. To the best of our knowledge, our solution CTDGC is the first graph condensation framework designed for event-stream continuous-time dynamic graphs. Unlike existing graph condensation methods for static or snapshot-based graphs, CTDGC directly operates on continuous event streams through a learnable selection-based condensation paradigm. Specifically, CTDGC learns event importance scores and constructs condensed graphs via class-wise top-k event selection. By leveraging a straight-through estimator, CTDGC enables end-to-end optimization of the non-differentiable top-k event selection process. By combining feature-level preservation and task-level preservation objectives, CTDGC effectively retains both temporal representations and task-relevant semantic information during condensation. Extensive experiments on multiple benchmark datasets demonstrate that CTDGC consistently outperforms competitive baselines under various compression ratios and generalizes well across different downstream temporal graph learning models. These results highlight the substantial redundancy in continuous-time event streams and validate the effectiveness of the proposed condensation framework. Future work will explore extending CTDGC to other continuous-time graph learning tasks and more advanced graph condensation settings.

AI USE STATEMENT

A large language model was used to assist with language polishing and manuscript editing. The authors verified all technical content, including the methodology, mathematical formulations, experimental settings, and reported results, and take full responsibility for the accuracy and integrity of this work.

ETHICS STATEMENT

This work studies graph condensation for continuous-time dynamic graphs using publicly available benchmark datasets. No human subjects were involved, and no personally identifiable information was collected or processed. The source code will be made publicly available upon acceptance of this paper.

REFERENCES

Chao Chen, Haoyu Geng, Nianzu Yang, Xiaokang Yang, and Junchi Yan. Easydgl: Encode, train and interpret for continuous-time dynamic graph learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(12):10845–10862, 2024.

Dong Chen, Shuai Zheng, Yeyu Yan, Muhao Xu, Zhenfeng Zhu, and Yao Zhao. Evolution-consistent dynamic graph condensation. *Pattern Recognit.*, 180:114169, 2026.

Weilin Cong, Si Zhang, Jian Kang, Baichuan Yuan, Hao Wu, Xin Zhou, Hanghang Tong, and Mehrdad Mahdavi. Do we really need complicated model architectures for temporal networks? In *ICLR*, 2023.

Marek Dedic and Michal Belohlávek. Benchmarking and transfer learning for hyperparameter optimization of graph neural networks. In *ICAART*, pp. 3079–3086, 2026.

Junfeng Fang, Xinglin Li, Yongduo Sui, Yuan Gao, Guibin Zhang, Kun Wang, Xiang Wang, and Xiangnan He. EXGC: bridging efficiency and explainability in graph condensation. In *WWW*, pp. 721–732. ACM, 2024.

Qizhang Feng, Zhimeng Stephen Jiang, Ruiquan Li, Yicheng Wang, Na Zou, Jiang Bian, and Xia Hu. Fair graph distillation. In *NeurIPS*, 2023.

ZhengZhao Feng, Rui Wang, TianXing Wang, Mingli Song, Sai Wu, and Shuibing He. A comprehensive survey of dynamic graph neural networks: Models, frameworks, benchmarks, experiments and challenges. *IEEE Trans. Knowl. Data Eng.*, 38(1):26–46, 2026.

Jian Gao, Jianshe Wu, and Jingyi Ding. Heterogeneous graph condensation. *IEEE Trans. Knowl. Data Eng.*, 36(7):3126–3138, 2024a.

Xinyi Gao, Tong Chen, Wentao Zhang, Yayong Li, Xiangguo Sun, and Hongzhi Yin. Graph condensation for open-world graph learning. In *ACM SIGKDD*, pp. 851–862, 2024b.

Xinyi Gao, Guanhua Ye, Tong Chen, Wentao Zhang, Junliang Yu, and Hongzhi Yin. Rethinking and accelerating graph condensation: A training-free approach with class partition. In *WWW*, pp. 4359–4373. ACM, 2025a.

Xinyi Gao, Junliang Yu, Tong Chen, Guanhua Ye, Wentao Zhang, and Hongzhi Yin. Graph condensation: A survey. *IEEE Trans. Knowl. Data Eng.*, 37(4):1819–1837, 2025b.

Peter J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35: 492–518, 1964.

Wei Jin, Xianfeng Tang, Haoming Jiang, Zheng Li, Danqing Zhang, Jiliang Tang, and Bing Yin. Condensing graphs via one-step gradient matching. In *ACM SIGKDD*, pp. 720–730, 2022.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *ACM SIGKDD*, pp. 1269–1278. ACM, 2019.

Yuxuan Liang, Wentao Zhang, Xinyi Gao, Ling Yang, Chong Chen, Hongzhi Yin, Yunhai Tong, and Bin Cui. Training-free heterogeneous graph condensation via data selection. In *ICDE*, pp. 1718–1731. IEEE, 2025.

Yang Liu, Deyu Bo, and Chuan Shi. Graph distillation with eigenbasis matching. In *ICML*, pp. 30702–30717, 2024a.

Zhanyu Liu, Chaolv Zeng, and Guanjie Zheng. Graph data condensation via self-expressive graph structure reconstruction. In *ACM SIGKDD*, pp. 1992–2002, 2024b.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *NeurIPS*, pp. 20596–20607, 2021.

Vipul Kumar Singh, Jyotismita Barman, Sandeep Kumar, Tapan K. Gandhi, and Jayadeva. Leap of FAITH from gnn-to-mlp: Fairness aware inference via distillation of graph knowledge. In *AAAI*, pp. 25490–25498, 2026.

Rakshit S. Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *ICLR*, 2019.

Lin Wang and Qing Li. Efficient graph condensation via gaussian process. *IEEE Trans. Knowl. Data Eng.*, 38(1):248–260, 2026.

Lu Wang, Xiaofu Chang, Shuang Li, Yunfei Chu, Hui Li, Wei Zhang, Xiaofeng He, Le Song, Jingren Zhou, and Hongxia Yang. TCL: transformer-based dynamic graph modelling via contrastive learning. *CoRR*, abs/2105.07944, 2021a.

Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. Inductive representation learning in temporal networks via causal anonymous walks. In *ICLR*, 2021b.

Zhenbang Xiao, Yu Wang, Shunyu Liu, Bingde Hu, Huiqiong Wang, Mingli Song, and Tongya Zheng. Disentangled condensation for large-scale graphs. In *WWW*, pp. 4494–4506. ACM, 2025.

Da Xu, Chuanwei Ruan, Evren Körpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. In *ICLR*, 2020.

Yuanyuan Xu, Wenjie Zhang, Ying Zhang, Xuemin Lin, and Xiwei Xu. Unlocking multi-modal potentials for link prediction on dynamic text-attributed graphs. In *AAAI*, pp. 27386–27394, 2026.

Zhe Xu, Yuzhong Chen, Menghai Pan, Huiyuan Chen, Mahashweta Das, Hao Yang, and Hanghang Tong. Kernel ridge regression-based graph dataset distillation. In *ACM SIGKDD*, pp. 2850–2861, 2023.

Yu Xue, Keyu Liu, Tan Yiyu, and Yong Zhang. Similarity surrogate-assisted evolutionary neural architecture search based on graph neural network. *Knowl. Based Syst.*, 337:115410, 2026.

Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley J. Osher, Yingyong Qi, and Jack Xin. Understanding straight-through estimator in training activation quantized neural nets. In *ICLR*, 2019.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. In *NeurIPS*, 2023.

Qiao Yuan, Sheng-Uei Guan, Pin Ni, Tianlun Luo, Prudence W. H. Wong, Victor Chang, and Ka Lok Man. Continual graph learning: A survey. *Pattern Recognit.*, 179:113600, 2026.

Jiasheng Zhang, Jialin Chen, Menglin Yang, Aosong Feng, Shuang Liang, Jie Shao, and Rex Ying. DTGB: A comprehensive benchmark for dynamic text-attributed graphs. In *NeurIPS*, 2024a.

Siwei Zhang, Xi Chen, Yun Xiong, Xixi Wu, Yao Zhang, Yongrui Fu, Yinglong Zhao, and Jiawei Zhang. Towards adaptive neighborhood for advancing temporal interaction graph modeling. In *ACM SIGKDD*, pp. 4290–4301. ACM, 2024b.

Yuan Zhang, Huawei Cao, Yue Shen, Sibo Li, Yang Feng, Chunxiang Wang, Xuejun An, and Xiaochun Ye. A comprehensive survey on dynamic graph processing: Storage and analytics. *IEEE Trans. Knowl. Data Eng.*, 38(5):2488–2507, 2026.

Yuchen Zhang, Tianle Zhang, Kai Wang, Ziyao Guo, Yuxuan Liang, Xavier Bresson, Wei Jin, and Yang You. Navigating complexity: Toward lossless graph condensation via expanding window matching. In *ICML*, pp. 60379–60395, 2024c.

Xin Zheng, Miao Zhang, Chunyang Chen, Quoc Viet Hung Nguyen, Xingquan Zhu, and Shirui Pan. Structure-free graph condensation: From large-scale graphs to condensed graph-free data. In *NeurIPS*, 2023.