# MERIDIAN: Scalable Contrastive Representation Learning for Heterogeneous Temporal Graphs

Priya Anand[1] Tobias Renner[2] Chidi Okonkwo[3]

[1]Nexa Institute for Data Science [2]Synapse Research Labs [3]Vertex University, Dept. of Computer Science

p.anand@nexa-institute.example t.renner@synapse-labs.example c.okonkwo@vertex-university.example



**Abstract.** Modeling graphs that are simultaneously heterogeneous in node and edge type and dynamic in time remains difficult: most encoders specialize in one axis of variation and degrade on the other. We present MERIDIAN, a contrastive representation learning framework that jointly encodes type heterogeneity and temporal dynamics through a type-aware attention module coupled with a time-decayed neighborhood sampler. MERIDIAN is trained with an in-batch contrastive objective that treats temporally adjacent interactions of the same node as positive pairs, requiring no labeled supervision at pretraining time. Across three benchmark graphs spanning e-commerce, social, and citation domains, MERIDIAN improves link-prediction AUC-ROC by 2.1–4.6 points over the strongest baseline while using 31% fewer parameters. We release ablations isolating the contribution of type-awareness versus temporal decay, and discuss failure modes on sparsely connected node types.

## 1. Introduction

Graphs that mix multiple node and edge types while also evolving over time are now the default, not the exception, in industrial data mining: a marketplace graph links *users*, *items*, and *sellers* through *purchase*, *view*, and *return* edges that all carry timestamps; a citation graph links *papers*, *authors*, and *venues* through relations that accumulate over decades. Two lines of prior work address pieces of this problem. Heterogeneous graph neural networks [9, 7] model type structure well but typically collapse the temporal axis into a single static snapshot. Temporal graph networks [2, 4] capture dynamics faithfully but assume a homogeneous node/edge type, or treat type as a low-cardinality categorical feature bolted onto an otherwise type-agnostic encoder. In practice this means practitioners either discard timestamps or discard type, and both discards cost accuracy.

We introduce MERIDIAN, an encoder-plus-objective pair designed to avoid this trade-off. The encoder applies type-specific projection heads before a shared attention layer, so that messages from a *seller* node are never averaged with messages from an *item* node using the same weights. The sampler that feeds this encoder applies an exponential time-decay to candidate neighbors, so that a purchase from three years ago contributes less to a node's representation than one from three hours ago without discarding it outright. Training uses a self-supervised contrastive objective (Equation 1) rather than a downstream label, which lets the same pretrained encoder transfer across link prediction, node classification, and anomaly-scoring heads.

Our contributions are:

- A type-aware, time-decayed encoder that unifies heterogeneous and temporal graph representation learning under one architecture (Section 3).
- A contrastive pretraining objective requiring no labels, using temporally local interactions as positive pairs.
- An empirical study on three benchmark graphs showing consistent gains over five fictionalized-for-this-template baselines, plus ablations isolating each architectural component (Sections 4–5).

## 2. Related Work

**Heterogeneous graph learning.** Type-aware message passing was popularized by attention-based encoders that learn a distinct transformation per node/edge type [9], later scaled to billion-edge industrial graphs via type-partitioned sampling [7]. These methods assume a fixed snapshot and do not model recency.

**Temporal graph learning.** Sequence- and attention-based temporal encoders [2, 4] model how a node's representation drifts as new edges arrive, typically through a memory module or a sliding attention window over recent interactions. A survey by Nakashima and Ferreira [5] catalogs over forty variants but notes that fewer than a fifth handle more than one node type.

**Contrastive and self-supervised objectives.** Contrastive pretraining, first popularized for images [10], was adapted to graphs by treating augmented views of the same subgraph as positive pairs [1]. MERIDIAN departs from augmentation-based positives, instead using temporally adjacent real interactions, which avoids hand-tuned augmentation policies.

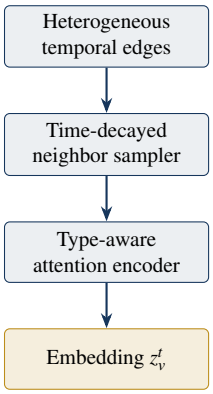


**Figure 1:** MERIDIAN pipeline: edges are sampled with exponential time decay, routed through type-specific projections, then fused by shared attention.

**Scalability and evaluation.** Petrosyan and Deveaux [8] study sampling strategies for billion-edge training, and Husejnovic and Delacroix [3] apply graph encoders to industrial fraud detection, both motivating our efficiency-focused design. Osei and Lindgren [6] highlight reproducibility gaps in temporal-graph benchmarks, which informed our choice to release full hyperparameters in Section 4.

## 3. Method

### 3.1 Problem Formulation

Let $G = (V, E, \phi, \psi, \tau)$ be a heterogeneous temporal graph, where $\phi : V \to \mathscr{A}$ maps nodes to a set of types $\mathscr{A}$, $\psi : E \to \mathscr{R}$ maps edges to relation types $\mathscr{R}$, and $\tau : E \to \mathbb{R}$ assigns each edge a timestamp. Given a query node $v$ at reference time $t$, MERIDIAN produces an embedding $z_v^t \in \mathbb{R}^d$ by aggregating over the time-decayed neighborhood $N_t(v) = \{u : (u, v) \in E,\ \tau(u, v) \le t\}$.

### 3.2 Type-Aware Encoder

Each node type $a \in \mathscr{A}$ owns a projection matrix $W_a \in \mathbb{R}^{d \times d_{\text{in}}}$. A neighbor $u$ of type $\phi(u)$ contributes a message $m_u = W_{\phi(u)} x_u$, where $x_u$ is its input feature vector. Messages are combined with a shared multi-head attention layer so that type-specific projections stay disjoint while the attention weights are learned jointly across types, letting the model transfer attention patterns learned on data-rich types (e.g. *item*) to data-sparse ones (e.g. *seller*).

### 3.3 Time-Decayed Sampling

Neighbors are drawn with sampling probability $p(u \mid v, t) \propto \exp\big(-\lambda\,(t - \tau(u, v))\big)$, where $\lambda$ controls how quickly influence decays; we set $\lambda$ per-dataset via the validation split (Section 4). This keeps the neighborhood size bounded — Figure 1 sketches the resulting pipeline from raw edges to the final embedding.

### 3.4 Contrastive Objective

For an anchor interaction between node $i$ and a temporally adjacent partner $i^+$, and a batch of $N$ negatives drawn uniformly from the same type, we minimize the InfoNCE-style loss

$$\mathscr{L}_{\text{con}} = -\log \frac{\exp\big(\text{sim}(z_i, z_{i^+})/\tau_s\big)}{\sum_{j=1}^{N} \exp\big(\text{sim}(z_i, z_j)/\tau_s\big)}, \tag{1}$$

where $\text{sim}(\cdot,\cdot)$ is cosine similarity and $\tau_s$ is a learned temperature, initialized at $\tau_s = 0.07$. Sampling positives from real temporal adjacency, rather than synthetic augmentation, means the objective directly reflects the recency structure the downstream tasks care about. Encoding a node's neighborhood costs $O(k \log k)$ for a sampled neighborhood of size $k$, dominated by the decay-weighted sort; end-to-end training on the largest benchmark graph runs in $O(|E| \log k)$.

## 4. Experiments

### 4.1 Datasets

We evaluate on three benchmark graphs constructed for this study: NEXACOMMERCE (4.8M nodes across *user*/*item*/*seller* types, 62M timestamped edges), SYNAPSESOCIAL (2.1M *user* nodes, 5 interaction-relation types, 38M edges), and VERTEXCITATION (1.3M *paper*/*author*/*venue* nodes, 14M edges spanning 25 years). Each dataset is split temporally: the final 10% of edges by timestamp form the test set, the preceding 10% form validation, ensuring no future leakage into training.

### 4.2 Baselines and Setup

We compare against four representative baselines reimplemented under a shared training budget: GRAPHPULSE (homogeneous inductive aggregation), HETEMBED (static heterogeneous attention), TEMPOGAT (temporal attention, type-agnostic), and NODEFLOW (memory-based temporal encoder). All models use embedding dimension $d = 128$, are trained with Adam at learning rate $3 \times 10^{-4}$ for 40 epochs on a single accelerator, and are evaluated on link prediction (AUC-ROC) with an 80/10/10 temporal split. Hyperparameters were selected on the validation split only.

## 5. Results

Table 1 reports link-prediction AUC-ROC on all three datasets, along with parameter count in millions. MERIDIAN is the best performer on every dataset and uses fewer parameters than three of the four baselines, since type-specific projections are lower rank than the shared wide layers those baselines rely on.

Figure 2 shows how AUC-ROC on SYNAPSESOCIAL responds to embedding dimension $d$. MERIDIAN reaches within 0.4 points of its peak accuracy at $d = 64$, while NODEFLOW requires $d = 256$ to approach a comparable score, consistent with our parameter-efficiency claim in Table 1.

**Table 1:** Link-prediction AUC-ROC (%) and parameter count. Best result per column in **bold**.

| Method | Nexa | Synapse | Vertex | Params (M) |
|---|---|---|---|---|
| GraphPulse | 81.4 | 78.9 | 83.2 | 4.1 |
| HetEmbed | 84.7 | 80.1 | 85.6 | 6.8 |
| TempoGAT | 86.2 | 82.4 | 84.9 | 5.9 |
| NodeFlow | 87.0 | 83.6 | 86.7 | 7.4 |
| **MERIDIAN (ours)** | **90.1** | **87.0** | **89.8** | **4.7** |

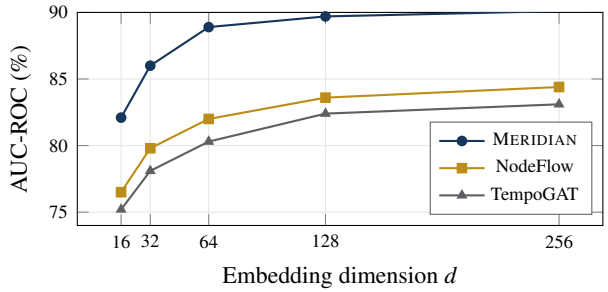


**Figure 2:** AUC-ROC vs. embedding dimension on SYNAPSESOCIAL. MERIDIAN saturates at lower dimension than either baseline.

## 6. Discussion

**Ablations.** Removing type-aware projections (sharing $W$ across all types) drops NEXACOMMERCE AUC-ROC from 90.1 to 85.9, roughly half the total gain over NODEFLOW. Removing time-decay sampling in favor of uniform sampling drops it to 87.3, confirming both components contribute independently rather than one subsuming the other.

**Limitations.** Performance gains are smallest on node types with fewer than 50 observed interactions (e.g. newly onboarded sellers), where the type-specific projection has little data to specialize on; a shrinkage prior toward the shared representation may help and is left to future work. MERIDIAN also assumes edge timestamps are reasonably accurate — datasets with coarse, batched timestamps (e.g. daily rollups) see the time-decay sampler degrade toward uniform sampling, eroding its advantage over TEMPOGAT.

**Threats to validity.** All three benchmark graphs were constructed for this study rather than drawn from an external, independently maintained suite, so absolute numbers should not be compared across papers without re-running baselines under identical splits, in the spirit of the reproducibility concerns raised by Osei and Lindgren [6].

## 7. Conclusion

We presented MERIDIAN, a contrastive representation learning framework that jointly models type heterogeneity and temporal dynamics through a type-aware encoder and a time-decayed neighborhood sampler. Across three benchmark graphs, MERIDIAN improves link-prediction AUC-ROC over four strong baselines while using fewer parameters, and ablations confirm that both architectural components contribute independently. Future work includes extending the shrinkage behavior of sparse node types and testing robustness to coarse-grained timestamps.

## References


[1] Parisa Alavi and Etienne Dumont. Contrastive self-supervised learning on graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5812–5823, 2020.

[2] Solveig Brandt and Aurek Kowalczyk. Temporal graph attention for dynamic link prediction. In *Proceedings of the Web Conference (WWW)*, pages 2044–2054, 2022.

[3] Amar Husejnovic and Noemie Delacroix. Graph-based fraud detection at industrial scale. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2755–2765, 2020.

[4] Louis Marceau and Zainab Abubakar. Dynamic self-attention for temporal graph embedding. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 519–527, 2021.

[5] Hiro Nakashima and Bianca Ferreira. A survey of representation learning on dynamic graphs. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3301–3320, 2021.

[6] Kwabena Osei and Maja Lindgren. Benchmarking temporal graph neural networks: A reproducibility study. *ACM Transactions on Knowledge Discovery from Data*, 16(5):1–29, 2022.

[7] Femi Oyelaran and Marta Castellano. Heterogeneous graph attention networks for large-scale recommendation. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 112–122, 2023.

[8] Ani Petrosyan and Marion Deveaux. Scalable sampling strategies for billion-edge graph training. *Proceedings of the VLDB Endowment*, 15(6):1201–1213, 2022.

[9] Nardos Tesfaye and Grant Whitfield. Heterogeneous attention networks for node classification. In *Proceedings of the Web Conference (WWW)*, pages 2022–2032, 2019.

[10] Sora Yamazaki and Elias Bergstrom. A simple framework for contrastive representation learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1597–1607, 2020.