FROM COMPLEX HISTORIES TO SIMPLE PREDICTORS: PREDICTIVE DIMENSION IN DYNAMIC GRAPH LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing continuous-time graph learning methods rely heavily on complex neural architectures to encode ever-expanding interaction histories. However, the future conditional distribution induced by complex histories can often be characterized by a low-dimensional predictive state manifold, implying that only a few predictive degrees of freedom are essential for future interactions. We therefore argue that continuous-time graph learning should identify low-dimensional states sufficient for future prediction rather than reconstruct the full history. To make this perspective measurable, we define and independently estimate the intrinsic predictive dimension, which quantifies the effective predictive degrees of freedom required by a given task. Specifically, we remove the history-independent marginal from the history-conditioned future distribution, decompose the resulting predictive residual operator, and estimate the minimum effective rank needed to preserve the predictive power of the full history. Experiments across datasets show that model performance saturation is largely determined by the intrinsic predictive dimension. Based on this finding, we propose PreDyG, a lightweight model whose capacity is matched to the intrinsic predictive dimension. PreDyG achieves superior predictive performance with only 0.07% of the parameters of a representative complex baseline. The code is available at https://anonymous.4open.science/r/PreDyG.

1 INTRODUCTION

Continuous-time graph learning aims to predict future interactions from expanding timestamped event streams Kazemi et al. (2020). Existing methods mostly treat interaction history as the primary modeling object, representing it through node memory, temporal message passing, recurrent updates, or Transformer encoders Rossi et al. (2020); Trivedi et al. (2019); Yu et al. (2023). Recent lightweight architectures suggest that such complex modeling mechanisms are not always necessary Cong et al. (2023); Tian et al. (2024). However, simplifying the architecture alone does not answer a more fundamental question: for a given prediction task, how many predictive degrees of freedom must a model actually retain?

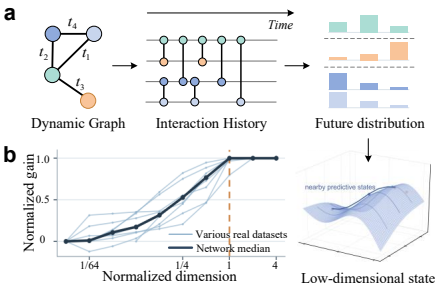


Figure 1: **(a)** Different histories can induce similar future distributions and nearby predictive states. **(b)** Predictive gain versus normalized dimension. Thin lines denote networks; the dark line shows the median; the dashed line marks dimension matching.

The key observation is that observational complexity is not the same as predictive complexity. Different histories may induce similar conditional distributions over future interactions and are therefore equivalent in the predictive sense Littman et al. (2001); Bialek (2022), as illustrated in Figure 1a. We find that these history-conditioned future distributions concentrate on a low-dimensional predictive-state manifold, meaning that only a small number of predictive degrees of freedom truly determine the future. This finding reframes continuous-time graph learning from reconstructing the full history to identifying a low-dimensional state that is sufficient for future prediction.

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.

We formalize this structure through the intrinsic predictive dimension. Given a prediction target and a tolerance level, the intrinsic predictive dimension is defined as the minimum predictive-state dimension required to retain a prescribed fraction of the achievable future-predictive gain. To estimate it independently of the final neural model, we first subtract the no-history marginal distribution from the history-conditioned future distribution, and then perform a low-rank decomposition of the resulting predictive residual operator. We then select, on held-out data, the minimum effective rank that approximately preserves the predictive ability of the full history. This separation makes predictive dimension a testable data-side quantity rather than an architecture-dependent hyperparameter. It also allows the predictive degrees of freedom required by a task to be measured before architecture design and independently of the final neural model.

Across diverse real continuous-time networks from social, communication, and mobility domains, we observe consistent low-dimensional predictive structure. The estimated predictive dimension is positively aligned with the saturation point of predictive gain and parameter budget, provides a more informative capacity ordering than node or event counts. After normalizing model dimension by the estimated predictive dimension, capacity curves become more consistent across datasets: model performance typically enters a plateau when the state dimension approaches the predictive dimension, whereas under-dimensioned models recover substantially less predictive gain. Figure 1b shows that, despite large differences in network scale, event count, and raw performance, predictive gains exhibit similar rising and saturation trends when normalized by intrinsic predictive dimension.

We further examine whether this relationship arises from genuine history–future dependencies or merely reflects statistical differences associated with dataset identity. To this end, we construct intervention experiments that preserve selected marginal statistics while separately disrupting temporal order, source-node identity, destination-node identity, or event pairing. These interventions induce coherent changes in the estimated predictive dimension and the independently measured saturation dimension, while also revealing dataset-specific responses to temporal and structural dependencies. This evidence argues against interpreting predictive dimension as a proxy for graph size, event count, dataset identity, or generic compressibility. Rather, it captures the effective predictive directions through which history constrains future interactions.

Based on this empirical regularity, we propose the Predictive Dimension Principle: within a fixed architecture family, model-state capacity should match the intrinsic predictive dimension of the interaction process, rather than the scale of the observed history. Guided by this principle, we design **Pre**dictive dimension-Guided **Dy**namic **G**raph Learning (**PreDyG**), an extremely compact architecture that uses low-dimensional states to model effective predictive degrees of freedom. Its compactness does not come from post-hoc pruning or model compression, but from allocating model capacity according to a data-side estimate of predictive complexity. Experiments compare the dimension-guided model with under-capacity models, over-capacity models, and conventional temporal graph architectures in terms of predictive accuracy, parameter count, training cost, and inference time. Our contributions are summarized as follows:

- We formalize intrinsic predictive dimension for continuous-time interaction streams and develop an estimator independent of the final neural model. This makes the predictive degrees of freedom required by a given task measurable before architecture design.
- We establish an empirical relationship between task-aligned predictive dimension and capacity saturation across diverse temporal networks. Predictive dimension indicates where additional capacity yields diminishing predictive gains, and experiments further connect this relationship to history–future dependencies.
- Guided by the Predictive Dimension Principle, we develop PreDyG, a compact temporal graph model. It achieves the best performance among the compared methods on most evaluated datasets across diverse domains, using only a significantly smaller number of trainable parameters compared to the baseline.

2 Related Work

Continuous-time graph learning. Continuous-time graph models encode timestamped histories through recurrent memory and message passing Kumar et al. (2019); Trivedi et al. (2019); Rossi et al. (2020), temporal neighborhoods and walks Xu et al. (2020); Wang et al. (2021b); Jin et al. (2022), or

mixing, Transformer, frequency-domain, and state-space operators Wang et al. (2021a); Cong et al. (2023); Yu et al. (2023); Tian et al. (2024); Gravina et al. (2024); Li et al. (2025). Complementary work studies model expressivity and execution efficiency Souza et al. (2022); Luo & Li (2022); Zhou et al. (2022); Lu et al. (2024). These studies mainly ask how history should be represented or processed. GraphMixer and CRAFT Cong et al. (2023); Yi et al. (2026) show that simplified temporal-graph architectures can remain competitive, but neither provides a data-side criterion for determining when a compact predictive state is sufficient.

Temporal graph complexity. Predictive-state representations define state by the future distributions it preserves Littman et al. (2001), while predictive dimension measures the past features needed for informative prediction Bialek (2022). Low-dimensional network geometry and shortest compression length provide complementary notions of structural complexity and predictability Almagro et al. (2022); Sun et al. (2020); temporal null models isolate genuine temporal dependence Gauvin et al. (2022). However, these lines neither estimate prediction-sufficient dimension for continuous-time graph events nor connect it to capacity saturation. We address both gaps by independently estimating predictive dimension and using it to guide model capacity.

3 Problem Formulation

Continuous-time interaction. Let $\mathcal{V}$ be the node set and $\mathcal{E} = \{e_k\}_{k=1}^{M}$ a chronologically ordered event stream, where $e_k = (u_k, v_k, t_k, \boldsymbol{x}_k)$ denotes an interaction from source $u_k \in \mathcal{V}$ to destination $v_k \in \mathcal{V}$ at time t_k, optionally with event features $\boldsymbol{x}_k$. The history immediately before t_k is $\mathcal{H}_k = \{e_j : t_j < t_k\}$. Given a query q_k (e.g., a source and timestamp), temporal link prediction estimates a future variable $Y_{k,h}$, such as the destination of the next interaction or the existence of an edge within horizon h.

Intrinsic predictive dimension. A state map g_d compresses the history into $Z_k = g_d(\mathcal{H}_k, q_k) \in \mathbb{R}^d$. Histories $\mathcal{H}$ and $\mathcal{H}'$ are predictively equivalent for target $Y_{k,h}$ when

$$p(Y_{k,h} \mid \mathcal{H}, q_k) = p(Y_{k,h} \mid \mathcal{H}', q_k). \tag{1}$$

where $p(\cdot \mid \cdot)$ denotes a conditional distribution, $Y_{k,h}$ is the prediction target at horizon h for query k, q_k is the query information, and $\mathcal{H}$ and $\mathcal{H}'$ are two candidate histories. The equivalence classes can be far fewer than the observed histories. Accordingly, graph size and history length are not, by themselves, measures of predictive structural complexity. For a fixed prediction target, horizon, representation family, loss ℓ, and tolerance ϵ, define the best attainable risk at dimension d as

$$\mathcal{R}_h(d) = \inf_{g_d, a} \mathbb{E}[\ell(a(g_d(\mathcal{H}_k, q_k)), Y_{k,h})] \tag{2}$$

where g_d maps the history and query to a state in $\mathbb{R}^d$, a maps that state to a prediction, ℓ is the task loss, and the expectation is over query–target pairs from the interaction process. Let $\mathcal{R}_0$ be the no-history risk and $\mathcal{R}_h^\star = \inf_d \mathcal{R}_h(d)$ the unrestricted reference risk. We define

$$d_{\mathrm{pred}}(\epsilon, h) = \min\left\{ d : \mathcal{R}_h(d) \le \mathcal{R}_h^\star + \epsilon\left(\mathcal{R}_0 - \mathcal{R}_h^\star\right) \right\}. \tag{3}$$

where $\epsilon \in [0, 1]$ is the tolerated fraction of the predictive gain not recovered by the state, and h specifies the future horizon. Thus d_{pred} measures the state dimension needed to recover a $1 - \epsilon$ fraction of the predictive gain supplied by history. It measures predictive structure, not irreducible uncertainty and not a globally minimum neural-network parameter count. Its relationship to neural parameter count is therefore empirical and is evaluated within a fixed model family and training protocol.

4 Methodology

Our methodology separates measurement from model construction. We first define a model-independent estimator of predictive dimension, then state the empirical capacity law it is used to test. Only after establishing this data-side quantity do we introduce the Predictive Dimension Principle and the PreDyG architecture.

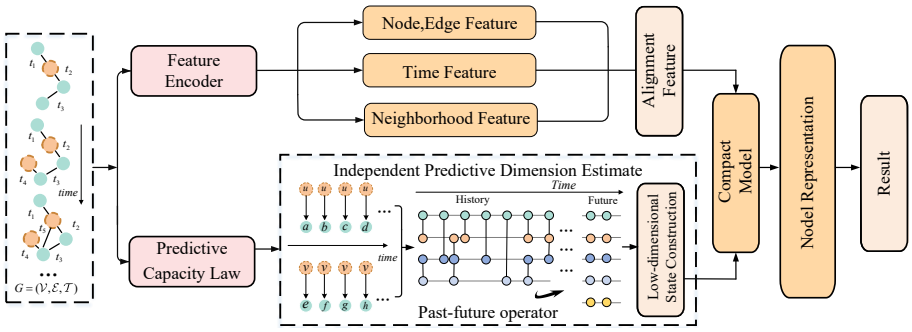


Figure 2: Overview of PreDyG, the encoder guided by intrinsic predictive dimension.

4.1 Independent Predictive Dimension Estimator

For finite event streams, estimating Equation 3 over unrestricted histories is ill posed. We construct a task-aligned future-distribution representation with ambient dimension D_{amb}. Source and destination identities are grouped using a fixed frequency-based rule, and the resulting representation is held identical across datasets. For event k, the context is

$$H_k = \big(b_s(u_k), b_d(v_{\mathrm{prev}(u_k)})\big), \qquad F_k = b_d(v_k), \tag{4}$$

where b_s and b_d are the fixed source- and destination-frequency binning maps, $v_{\mathrm{prev}(u_k)}$ is the most recent destination previously contacted by u_k, H_k is the discrete past context, and F_k is the one-step future label. This estimator is held identical across domains; it is not tuned to PreDyG.

Past–future operator. We estimate Q from event counts with additive 0.5 smoothing and center its history-dependent component:

$$C_{hf} = \sqrt{p_h}\big(Q_{hf} - \pi_f\big). \tag{5}$$

where h and f index values of H and F, $Q_{hf} = p(F = f \mid H = h)$ is a conditional probability, $p_h = p(H = h)$ is the context marginal, $\pi_f = \sum_h p_h Q_{hf}$ is the no-history future marginal, and C is the centered, marginal-weighted past–future operator. The marginal π is the one-dimensional no-history component. For a candidate dimension d, we retain $d-1$ orthogonal predictive coordinates of C, computed by truncated singular-value decomposition, reconstruct $Q^{(d)}$, clip negative entries, and normalize rows to the probability simplex. The singular values are not used directly as a complexity statistic; the predictive dimension is selected by held-out Brier risk.

Predictive dimension selection. We evaluate a logarithmically spaced rank grid up to D_{amb}. The first 40% of events fit the operator and the next 20% form the selection split. With $R^* = \min_d R(d)$ and $\epsilon = 0.05$, the predictive-dimension estimate is

$$\widehat{d}_{\mathrm{pred}} = \min\{d : R(d) \le R^* + 0.05[R(1) - R^*]\}. \tag{6}$$

where $R(d)$ is selection-split Brier risk at dimension d, $R^* = \min_d R(d)$ is the best risk among candidate dimensions, and $R(1)$ is the no-history reference risk; 0.05 fixes the tolerated loss of historical predictive gain. The first 70% are then refit and the last 30% independently determine the held-out predictive-rank estimate d_{hold} using the same 5% rule. This chronology lets $\widehat{d}_{\mathrm{pred}}$ predict, rather than define, d_{hold}.

4.2 Empirical Capacity Law

To compare predictive saturation across datasets, we normalize the history-dependent risk reduction as

$$G_D(d) = \frac{R_D(1) - R_D(d)}{R_D(1) - \min_j R_D(j)}. \tag{7}$$

where $R_D(d)$ is the held-out risk at predictive-state dimension d, $R_D(1)$ is the no-history risk, and j ranges over the candidate dimensions. When historical gain is positive, $G_D(d) = 1$ denotes the best gain observed in the dimension scan. A valid predictive-dimension estimate should organize these curves, with predictive gain approaching its plateau as $d/\widehat{d}_{\mathrm{pred}}$ approaches one.

4.3 PREDICTIVE DIMENSION PRINCIPLE

The empirical capacity law motivates the following design rule:

> *For a fixed prediction target, horizon, and tolerance, allocate model-state capacity according to the independently estimated intrinsic predictive dimension of the interaction process, rather than according to its number of nodes, events, or historical observations.*

This principle treats predictive dimension as a data-side scale for selecting a sufficient trainable-parameter budget. Within a fixed PreDyG architecture family, trainable capacity is jointly distributed across the predictive state, modality projections, fusion module, and task-specific readout. We therefore use $\widehat{d}_{\mathrm{pred}}$ to identify the parameter regime in which predictive performance saturates.

4.4 PREDYG: A PREDICTIVE-DIMENSION GUIDED ENCODER

we proposed PreDyG which implements the principle by parameterizing a low-dimensional predictive state, using explicit interaction statistics to expose recurring structure, and restricting learnable capacity to the components needed for future prediction.

Interaction-history features. For a query edge (u, v, t), retrieves the L most recent interactions of each endpoint. For neighbor w_j at time t_j, four inputs are constructed: its node feature, the edge feature, a fixed sinusoidal encoding of $t - t_j$, and a neighborhood-interaction-frequency (NIF) statistic. For endpoint pair (u, v), the latter records how often w_j appears in the local histories of u and v,

$$\boldsymbol{c}_{u,v}(w_j) = \left[\sum_{a \in \mathcal{N}_L(u,t)} \mathbf{1}[a = w_j], \sum_{a \in \mathcal{N}_L(v,t)} \mathbf{1}[a = w_j] \right], \tag{8}$$

where $\mathcal{N}_L(x, t)$ is the multiset of neighbors in endpoint x's L most recent pre-t interactions, a indexes a neighbor occurrence, $\mathbf{1}[\cdot]$ is the indicator function, and $\boldsymbol{c}_{u,v}(w_j) \in \mathbb{R}^2$ is the NIF count vector for neighbor w_j. This statistic exposes repeated and cross-neighborhood interaction structure without a learned node-sized memory table.

Low-dimensional state construction. Each modality is projected to a width-m channel and fused:

$$\boldsymbol{z}_j = W_r \big[W_x \boldsymbol{x}_{w_j} \,\|\, W_e \boldsymbol{e}_j \,\|\, W_t \boldsymbol{\tau}(t - t_j) \,\|\, W_n \phi(\boldsymbol{c}_{u,v}(w_j)) \big], \tag{9}$$

where $\boldsymbol{x}_{w_j}$ and $\boldsymbol{e}_j$ are the neighbor-node and event features, $\boldsymbol{\tau}$ is the fixed sinusoidal elapsed-time encoding, ϕ is the NIF encoder, W_x, W_e, W_t, W_n are modality projections, W_r is the fusion projection, $\|$ denotes concatenation, and $\boldsymbol{z}_j$ is the fused historical interaction. The compact instantiation combines three summaries of the historical interaction representations, namely their mean, coordinatewise maximum, and most recent representation, through a single projection:

$$\boldsymbol{h}_u(t) = W_p \big[\operatorname{mean}_j \boldsymbol{z}_j \,\|\, \max_j \boldsymbol{z}_j \,\|\, \boldsymbol{z}_L \big], \tag{10}$$

where $\boldsymbol{h}_u(t)$ is the predictive state of u, W_p is the pooling projection, the mean and maximum are computed coordinatewise over the L historical interaction representations, and $\boldsymbol{z}_L$ denotes the most recent representation. The same encoder produces $\boldsymbol{h}_v(t)$, and a linear pair readout gives

$$\hat{p}(u, v, t) = \sigma(\boldsymbol{w}_s^\top [\boldsymbol{h}_u(t) \| \boldsymbol{h}_v(t)]), \tag{11}$$

where $\hat{p}(u, v, t)$ is the predicted interaction probability, σ is the logistic sigmoid, and $\boldsymbol{w}_s$ is the learned pair-readout vector. Training uses binary cross-entropy with sampled negative destinations. The model has no recurrent node memory, temporal message-passing stack, or Transformer stack.

Dimension-guided capacity. For a fixed PreDyG architecture family, we use the task-aligned predictive dimension to identify a sufficient trainable-parameter budget. Specifically, we select the smallest configuration whose validation performance reaches at least 95% of the best result in the capacity scan. Predictive dimension therefore guides the total parameter budget, it is distributed across the predictive state, modality projections, fusion module, and readout. In general, the trainable parameter count scales as $O(m(p_x + p_e + p_t + p_n) + m^2 + p_n^2)$, where m is the predictive-state width and p_x, p_e, p_t, p_n are the input widths of the node, edge, time, and NIF channels.

5 EXPERIMENTS

We organize the experiments in two stages. Stage I estimates task-aligned predictive dimension and tests whether it organizes predictive-gain saturation across datasets. Stage II examines whether this data-side scale can guide the sufficient parameter budget of PreDyG, and then evaluates its predictive performance and efficiency. **Data-side measurement.** We study 12 continuous-time interaction streams spanning communication, proximity, politics, social behavior, recommender systems, and mobility, using at most the first 200,000 events of each stream. All datasets share the same prediction target and history representation, with Brier risk and a 5% tolerance. Predictive dimension is estimated on chronological data splits before neural-model training. Estimation details are provided in Appendix A.1; comparisons with scale and compression statistics are reported in Appendix A.3. **Model-side validation.** Dynamic link prediction uses chronological $70\%/15\%/15\%$ train/validation/test splits, with evaluation in transductive and inductive settings. We report average precision (AP) as mean $\pm$ standard deviation across repeated runs. We conducted experiments on 8 widely diverse datasets, and also set up 4 large TGB datasets following their official data splitting method and MRR evaluation protocol Huang et al. (2023). For the capacity analysis, we define the sufficient parameter budget $P_{0.05}(D)$ as the smallest trainable parameter count whose validation AP reaches at least 95% of the best value observed in the capacity scan. Datasets and baseline methods are described in Appendices D.1 and D.2, respectively.

5.1 PREDICTIVE DIMENSION ANALYSIS

5.1.1 PREDICTIVE DIMENSION AS A CAPACITY SCALE

We first test whether the future distributions induced by interaction histories occupy a compact predictive geometry. For each history context, we estimate its conditional future distribution and measure the local intrinsic dimension of these predictive fingerprints. As a reference, we use an independence null that preserves observation counts and future-event marginals while removing history–future dependence. This comparison asks whether the observed low-dimensionality reflects genuine predictive structure rather than marginal event statistics. Figure 3(a) shows that the estimated local dimension is only a small fraction of the ambient future-distribution dimension and is consistently below the independence-null reference. This indicates that histories do not scatter future distributions arbitrarily in the ambient space; instead, they induce a compact set of reusable predictive directions.

We next ask whether this low-dimensional structure is relevant to model capacity. Figure 3(b) compares the independently estimated predictive dimension $\widehat{d}_{\mathrm{pred}}$ with the held-out dimension d_{hold}. The two estimates coincide on most streams and remain close on the discrete dimension grid, suggesting that $\widehat{d}_{\mathrm{pred}}$ identifies the capacity scale at which additional predictive-state dimensions provide diminishing returns.

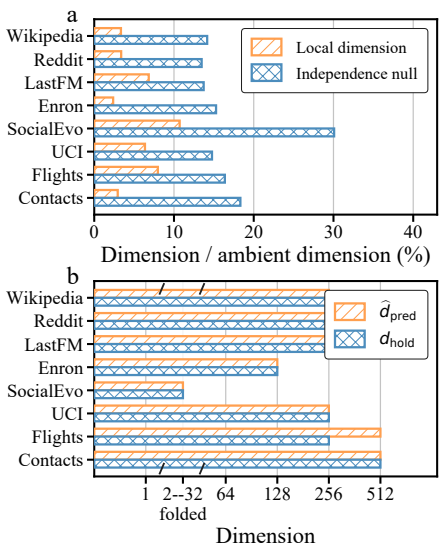


Figure 3: (a) History-conditioned future distributions are lower-dimensional than the independence null. (b) The estimated predictive dimension $\widehat{d}_{\mathrm{pred}}$ aligns with the held-out saturation dimension d_{hold}, marking the onset of capacity saturation.

5.1.2 PREDICTIVE DIMENSION AS A DEPENDENCY MEASURE

To test whether predictive dimension reflects genuine history–future dependencies rather than dataset identity or marginal event statistics, we construct dependency interventions that preserve event count while perturbing temporal order, destination identity, source identity, or source–destination pairs. We apply these interventions to first 80,000-event of Reddit, SocialEvo, and Contacts, and to the complete UCI stream. These streams span different predictive-dimension regimes and dependency profiles, while the first 80,000-event keeps the intervention budget comparable across larger datasets;

UCI is used in full because it is shorter. For each stochastic intervention, we report the median over three independently shuffled realizations.

As Table 1 shows, the two independently selected dimensions move concordantly across the 20 dataset–intervention pairs, but the dependency carrying predictive dimension differs by dataset. Perturbing destination or source identity reduces all four streams to, or close to, the one-dimensional floor. Temporal and pair perturbations instead produce dataset-specific responses: UCI falls from $256/256$ to $1/1$ under either perturbation, SocialEvo retains $32/16$ after temporal shuffling and falls to $16/16$ after pair shuffling, whereas Reddit and Contacts retain their original dimensions. Thus, predictive dimension responds selectively to the temporal and structural dependencies that constrain future interactions, rather than to event count, arbitrary randomization, or a fixed dataset label.

Together, these data-side analyses show that, for a fixed prediction target, tolerance, and estimator family, predictive gain generally saturates when the available state dimension reaches the independently estimated predictive dimension. The above results all connect $\widehat{d}_{\mathrm{pred}}$ to future-predictive degrees of freedom rather than observation scale or dataset identity.

5.2 From Predictive Dimension to Model Design

5.2.1 Predictive Dimension Guides Capacity Selection

We next test whether predictive dimension can guide a practical capacity choice. For each dataset, we keep the prediction task, data split, and training procedure fixed, and evaluate an ordered family of PreDyG configurations. We define the sufficient parameter budget as the smallest configuration that reaches at least 95% of the best value observed in the capacity scan. Figure 4(a) reveals a consistent transition from under-capacity to saturation across four representative datasets. Test AP improves sharply in the low-capacity regime but changes only marginally beyond the validation-selected $P_{0.05}$. The selected configuration therefore captures the transition from under-capacity to diminishing returns, rather than simply favoring the smallest model. Figure 4(b) extends this comparison to eight datasets. The sufficient parameter budget generally increases with relative predictive dimension: datasets with limited predictive structure saturate with smaller models, while those with richer predictive structure require more parameters. The relationship is not an exact conversion from dimension to parameter count, but it provides a data-dependent capacity scale before selecting the final architecture.

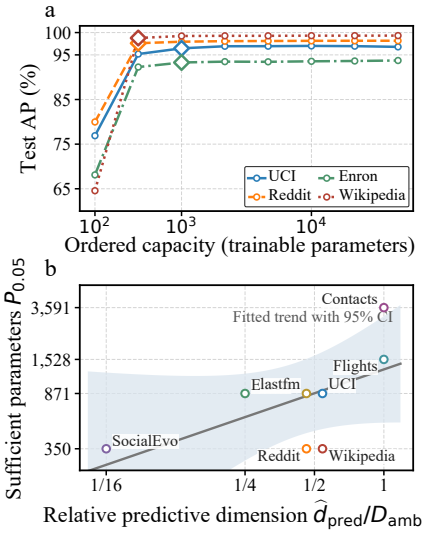


Figure 4: (a) Test AP across ordered capacity configurations; diamonds mark $P_{0.05}$. (b) $P_{0.05}$ versus $\widehat{d}_{\mathrm{pred}}/D_{\mathrm{amb}}$; the line and shading show the fitted trend and its 95% confidence interval.

This guidance also translates into end-to-end efficiency. The selected PreDyG configuration uses average about $1300\times$ fewer trainable parameters than the representative baseline while maintaining comparable transductive and new-node prediction performance. Predictive dimension therefore turns the data-side law into an actionable model-selection principle.

5.2.2 Efficiency Analysis

To evaluate whether the compact capacity regime also yields stable optimization and practical savings, we first compare validation trajectories across four interaction domains. As shown in Figure 5, PreDyG reaches a strong plateau rapidly on Wikipedia, and it stabilizes within the first few epochs on UCI and Enron. The behavior is consistent across datasets with different AP ranges and convergence horizons. Compactness therefore does not require unusually slow or unstable optimization. More comparison results are shown in Appendix D.3.

We further compare trainable parameters and measured time per epoch under the common benchmark implementation. As shown in Figure 6 (a), GraphMixer, TGN, and DyGFormer use on average $716\times$,

Table 1: Dependency interventions. Each cell reports $\widehat{d}_{\text{pred}}/d_{\text{hold}}$, followed by the percentage of the original predictive dimension retained.

Dataset	Original	Time/order	Destination	Source	Source–destination pair
Reddit	256/256 (100%)	256/256 (100%)	1/1 (< 1%)	1/1 (< 1%)	256/256 (100%)
SocialEvo	32/32 (100%)	32/16 (100%)	1/1 (3%)	1/1 (3%)	16/16 (50%)
UCI	256/256 (100%)	1/1 (< 1%)	1/1 (< 1%)	1/1 (< 1%)	1/1 (< 1%)
Contacts	512/512 (100%)	512/512 (100%)	1/1 (< 1%)	4/1 (1%)	512/512 (100%)

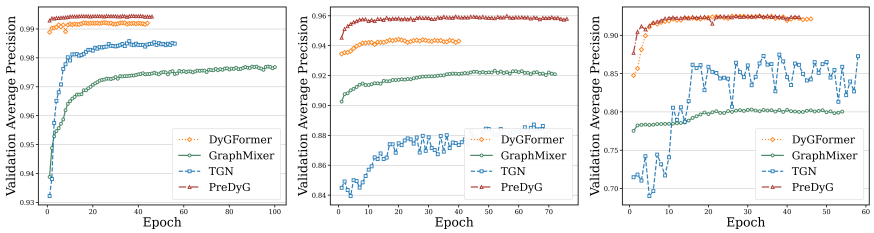


Figure 5: The AP metrics and generalization gap on multiple datasets with the early stopping mechanism were compared: (a) Wikipedia; (b) UCI; (c) Enron.

1,074×, and 1,314× as many trainable parameters as PreDyG, respectively. across dataset-specific feature configurations. Figure 6 (b) and (c) show the running time per epoch of the PreDyG algorithm compared with other baselines. Due to space limitations, we only present a portion of the results. As shown in the figure, we found that in the Wikipedia and UCI datasets, the running time of our algorithm PreDyG was significantly shorter than that of other algorithms, achieving higher validation AP with shorter per-round running time. These results show a favorable accuracy–efficiency trade-off for the evaluated implementation. More results and details can be found in Appendix C.

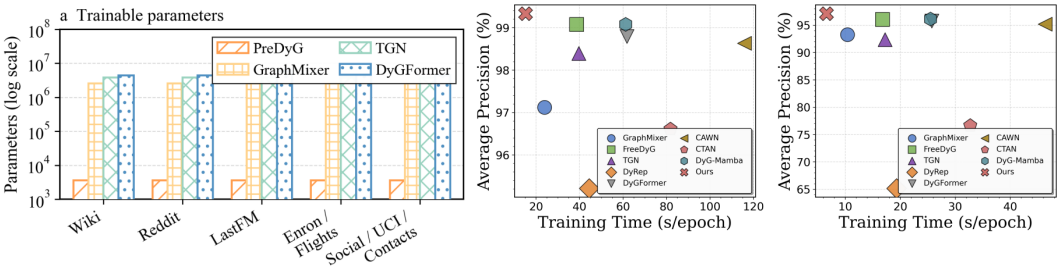


Figure 6: Training efficiency. (a) Trainable parameter counts, with baselines grouped into five comparable scales. Validation accuracy versus time per epoch on (b) Wikipedia and (c) UCI.

5.2.3 Cross-domain prediction performance

To test whether our proposed model can maintain excellent prediction quality across a wide range of real-world scenarios, we evaluated it on real-world datasets across different interaction domains with a very small number of parameters and low runtime costs. As shown in Table 2, our algorithm, PreDyG, achieved state-of-the-art performance on datasets from eight different domains. Furthermore, more complete results regarding the inductive approach can be found in Appendix D.4, where our algorithm also performed exceptionally well in predicting new nodes, achieving the best results. The superior performance in both settings demonstrates that low dimensionality is not limited to a single interaction domain. This is because different domains change the specific content of the predicted state, and the inherent dimensionality required for predicting the state does indeed exist across a broad range of domains.

To test the performance of PreDyG on much larger time-graph datasets, we further conducted experiments using TGB. The results are shown in Table 3, where PreDyG achieved the highest MRR on four large time-graph datasets, including tgbl-review and the tgbl-coin data stream with millions of nodes. This demonstrates that in more complex and longer-term temporal interactions, there still exists an intrinsic dimension required for predicting states, and it plays an important role.

5.2.4 Ablation Study

Table 2: Transductive average precision (%) on eight domains, reported as mean ± standard deviation. Best and second-best entries are bold and underlined, respectively.

Method	Wiki	Reddit	LastFM	Enron	SocialEvo	UCI	Flights	Contacts
JODIE	96.50±.18	98.31±.12	70.85±2.11	84.77±.28	89.89±.52	89.43±1.06	95.60±1.72	95.31±1.36
DyRep	94.86±.04	98.22±.07	71.92±2.20	82.38±3.32	88.87±.31	65.14±2.33	95.29±.69	95.98±.15
TGN	98.45±.05	98.63±.05	77.07±3.99	86.53±1.14	93.57±.21	92.34±1.02	97.95±.16	96.89±.54
CAWN	98.76±.03	99.11±.01	86.99±.07	89.56±.13	84.96±.09	95.18±.05	<u>98.51±.01</u>	90.26±.30
GraphMixer	97.25±.03	97.31±.01	75.61±.22	82.25±.16	93.37±.07	93.25±.54	90.99±.08	91.92±.03
DyGFormer	98.82±.02	99.11±.01	93.00±.12	92.47±.15	94.47±.01	95.76±.16	98.25±.01	<u>98.29±.01</u>
FreeDyG	<u>99.23±.01</u>	99.21±.01	90.74±.14	92.51±.05	<u>94.54±.01</u>	96.03±.02	98.20±.01	98.00±.01
CTAN	96.61±.79	97.21±.84	86.44±.80	92.52±1.20	94.32±.01	76.64±4.11	97.24±.01	96.29±.03
DyG-Mamba	99.08±.09	<u>99.12±.00</u>	<u>93.57±.01</u>	<u>92.34±.12</u>	94.50±.01	<u>96.14±.14</u>	98.03±.02	98.25±.01
PreDyG	**99.30±.01**	**99.34±.01**	**93.97±.06**	**93.18±.07**	**94.68±.01**	**97.01±.03**	**98.79±.01**	**98.32±.01**

Table 3: MRR (%) on large TGB datasets. Dashes denote unavailable reported results. Best and second-best entries are bold and underlined, respectively.

Method	tgbl-wiki	tgbl-review	tgbl-coin	tgbl-comment
CAWN	73.04±.60	19.30±.10	–	–
DyRep	51.91±1.95	<u>40.06±.59</u>	45.20±4.60	28.90±3.30
DyGFormer	77.36±.31	22.39±1.52	<u>75.17±.38</u>	67.03±.14
TCL	<u>78.11±.20</u>	16.51±1.85	68.66±.30	<u>70.11±.83</u>
TGAT	59.94±1.63	19.64±.23	60.92±.57	56.20±2.11
TGN	68.93±.53	37.48±.23	58.60±3.70	37.90±2.00
PreDyG	**78.79±.38**	**40.30±.46**	**76.21±.33**	**71.21±.72**

To identify the essential components of PreDyG and test whether its performance depends on a particular neighborhood-statistics construction, we conduct ablation experiments. As shown in Table 4, removing NIF causes the largest performance loss on every dataset, including a 9.13-point AP drop on Enron. This indicates that PreDyG relies on explicit, low-cost interaction statistics rather than hidden width alone. Removing the node, edge, and temporal modalities, or replacing the learned fusion module, also consistently reduces AP. Table 5 varies the construction of neighborhood interaction features. Path-NIF and Degree-NIF remain close to the default construction, whereas learned enumeration is less stable. Thus, the result does not depend on a single hand-designed enumeration rule, although the default NIF construction gives the best average balance.

Table 4: The results of Component ablation experiments.

Variant	Wiki	Reddit	Enron	UCI
Full PreDyG	**99.30**	**99.34**	**93.18**	**97.01**
w/o NIF	97.44	97.87	84.05	93.28
w/o modalities	98.16	98.73	90.99	95.19
w/o fusion	99.07	99.18	90.43	95.14

Table 5: The results of NIF sensitivity experiments.

Construction	Wiki	Reddit	Enron	UCI
Default NIF	**99.30**	**99.34**	**93.18**	**97.01**
Learned enum.	98.00	98.21	91.88	94.30
Path-NIF	99.27	99.12	91.90	96.86
Degree-NIF	99.34	99.20	93.12	96.90

6 Conclusion

This work estimates the prediction degrees independently of the final neural network model based on the concept of intrinsic prediction dimension. In various continuous-time networks, the task-aligned predictive dimension provides a more informative ordering of sufficient parameter budgets than graph scale. These findings prompt the proposal of the ”prediction dimension principle”: within a fixed architecture family, the capacity of the model state should match the degrees of freedom of prediction relevance, rather than the observed historical quantities. PreDyG implements this principle through a lightweight prediction state encoder, achieving performance comparable to or exceeding that of larger-scale models in different evaluation domains. Its compactness stems from the measured prediction structure, whose parameter regime is selected using the predictive-dimension analysis.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing, grammar correction, and wording refinement of the manuscript. We have not used generative AI tools for generating research ideas, formulating hypotheses, conducting experiments, and other required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for improving readability and clarity of presentation. We have reviewed all AI-assisted work. All AI-assisted edits were checked and revised by the authors to ensure technical correctness, consistency with the experimental results, and absence of unsupported claims. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Pedro Almagro, Marián Boguñá, and M. Ángeles Serrano. Detecting the ultra low dimensionality of real networks. *Nature Communications*, 13:6096, 2022. doi: 10.1038/s41467-022-33685-z.

William Bialek. On the dimensionality of behavior. *Proceedings of the National Academy of Sciences*, 119(18):e2021860119, 2022. doi: 10.1073/pnas.2021860119.

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 *International Conference on Learning Representations*, 2023.

Laetitia Gauvin, Mathieu Génois, Márton Karsai, Mikko Kivelä, Taro Takaguchi, Eugenio Valdano, and Christian L. Vestergaard. Randomized reference models for temporal networks. *SIAM Review*, 64(4):763–830, 2022. doi: 10.1137/19M1242252.

Alessio Gravina, Giulio Lovisotto, Claudio Gallicchio, Davide Bacciu, and Claas Grohnfeldt. Long range propagation on continuous-time dynamic graphs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16206–16225, 2024.

Shenyang Huang, Farimah Poursafaei, Jacob Danovitch, Matthias Fey, Weihua Hu, Emanuele Rossi, Jure Leskovec, Michael Bronstein, Guillaume Rabusseau, and Reihaneh Rabbany. Temporal graph benchmark for machine learning on temporal graphs. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, volume 36, 2023. doi: 10.52202/075280-0099.

Ming Jin, Yuan-Fang Li, and Shirui Pan. Neural temporal walks: Motif-aware representation learning on continuous-time dynamic graphs. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1445.

Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupart. Representation learning for dynamic graphs: A survey. *Journal of Machine Learning Research*, 21(70):1–73, 2020.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1269–1278, 2019. doi: 10.1145/3292500.3330895.

Dongyuan Li, Shiyin Tan, Ying Zhang, Ming Jin, Shirui Pan, Manabu Okumura, and Renhe Jiang. DyG-Mamba: Continuous state space modeling on dynamic graphs. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4299.

Michael L. Littman, Richard S. Sutton, and Satinder Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems*, volume 14, pp. 1555–1561, 2001.

Xiaodong Lu, Leilei Sun, Tongyu Zhu, and Weifeng Lv. Improving temporal link prediction via temporal walk matrix projection. *Advances in Neural Information Processing Systems*, 37:141153–141182, 2024. doi: 10.52202/079017-4482.

Yuhong Luo and Pan Li. Neighborhood-aware scalable temporal network representation learning. In *Proceedings of the First Learning on Graphs Conference*, volume 198 of *Proceedings of Machine Learning Research*, pp. 1:1–1:18, 2022.

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles E. Leiserson. EvolveGCN: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5363–5370, 2020. doi: 10.1609/aaai.v34i04.5984.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael M. Bronstein. Temporal graph networks for deep learning on dynamic graphs. In *ICML Workshop on Graph Representation Learning and Beyond*, 2020.

Amauri H. Souza, Diego Mesquita, Samuel Kaski, and Vikas Garg. Provably expressive temporal graph networks. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-2337.

Jiachen Sun, Ling Feng, Jiarong Xie, Xiao Ma, Dashun Wang, and Yanqing Hu. Revealing the predictability of intrinsic structure in complex networks. *Nature Communications*, 11:574, 2020. doi: 10.1038/s41467-020-14418-6.

Yadong Sun, Xiaofeng Cao, Ivor Tsang, and Heng Tao Shen. Analytical construction on geometric architectures: Transitioning from static to temporal link prediction. In *Forty-second International Conference on Machine Learning*.

Yuxing Tian, Yiyan Qi, and Fan Guo. FreeDyG: Frequency enhanced continuous-time dynamic graph model for link prediction. In *International Conference on Learning Representations*, 2024.

Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. DyRep: Learning representations over dynamic graphs. In *International Conference on Learning Representations*, 2019.

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. *arXiv preprint arXiv: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 *International Conference on Learning Representations*, 2021b.

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

Lu Yi, Runlin Lei, Fengran Mo, Yanping Zheng, Zhewei Wei, and Yuhang Ye. Future link prediction without memory or aggregation. *Advances in Neural Information Processing Systems*, 38:2347–2375, 2026.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. In *Advances in Neural Information Processing Systems*, volume 36, pp. 67686–67700, 2023. doi: 10.52202/075280-2960.

Hongkuan Zhou, Da Zheng, Israt Nisa, Vasileios Ioannidis, Xiang Song, and George Karypis. TGL: A general framework for temporal GNN training on billion-scale graphs. *Proceedings of the VLDB Endowment*, 15(8):1572–1580, 2022. doi: 10.14778/3529337.3529342.

A Implementation and Reproducibility Details

A.1 Predictive Dimension Estimation Details

History representation and prediction target. Events are sorted by timestamp, preserving their original order in the case of ties, and each stream is limited to its first 200,000 events. Source and destination identities are grouped separately by frequency using a fixed 512-bin rule. Up to 511 frequent identities receive individual bins and the remaining identities share a tail bin, giving the future-distribution representation an ambient dimension of $D_{\mathrm{amb}} = 512$. The prediction target is the destination bin of the current event. Its history representation consists of the current source bin and the destination bin of the most recent event involving that source; a separate symbol denotes missing history. The history representation is formed before the current interaction updates the event record.

Conditional distributions and low-rank approximation. Conditional probabilities are estimated from history–target counts in the fitting interval using additive smoothing with coefficient 0.5. History categories absent from this interval are assigned the smoothed destination marginal when constructing the conditional-probability table. We subtract this marginal from the conditional distributions, weight each row by the square root of its empirical history probability, and apply singular value decomposition to the resulting predictive residual matrix. For total dimension d, reconstruction retains $d-1$ residual directions and adds back the marginal; $d = 1$ therefore represents prediction without history. Reconstructed probabilities are clipped below at 10^{-8} and normalized within each row. The history-probability floor used in weighting and inverse weighting is 10^{-12}.

Risk and dimension selection. We evaluate each dimension using the mean Brier risk,

$$R(d) = \frac{1}{n} \sum_{k=1}^{n} \sum_{f=1}^{K} \left[q_d(f \mid H_k) - \mathbf{1}\{F_k = f\} \right]^2, \tag{12}$$

where n is the number of evaluation events, K is the number of destination categories, H_k is the history representation before event k, F_k is its destination category, q_d is the predictive distribution at dimension d, and $\mathbf{1}\{\cdot\}$ denotes the indicator function. Within a candidate set, the selected dimension is

$$d^\star = \min \left\{ d \in \mathcal{D} : R(d) - R_{\min} \le 0.05 \left[R(1) - R_{\min} \right] \right\}, \qquad R_{\min} = \min_{d \in \mathcal{D}} R(d), \tag{13}$$

where $\mathcal{D}$ is the candidate dimension set, $R(1)$ is the no-history risk, $R_{\min}$ is the lowest risk attained within that set, and $d^\star$ is the smallest dimension retaining at least 95% of the attainable risk reduction. If this reduction is at most 10^{-9}, the selected dimension is set to 1.

The first 40% of events fit the past–future operator and the next 20% select $\widehat{d}_{\mathrm{pred}}$ on the grid $\{1, 2, 4, 8, 16, 32, 64, 128, 256, 512\}$. We then refit the operator on the first 70% of events and apply the same grid and 5% risk criterion to the final 30%, producing the independent held-out estimate d_{hold}. Both estimates therefore use the same prediction target, ambient representation, rank grid, and tolerance, but are selected on disjoint chronological evaluation intervals.

A.2 Scale and Compression Statistics

Scale statistics. All competing statistics are computed on the same event prefix used for predictive-dimension estimation. Event count is the number of interactions in this prefix; source and destination counts enumerate distinct identities on their respective sides. Distinct-pair count enumerates directed source–destination pairs, counting repeated interactions only once. The node-scale correlation reported by the implementation uses the number of distinct source identities.

Event-stream encoding. Source and destination identities are relabeled separately in order of first occurrence. Each event is serialized as its source identifier, destination identifier, and time since the preceding event, with comma-separated fields and one event per line in a UTF-8 byte stream. Time differences are measured in the dataset's original timestamp units, clipped below at zero, and rounded down to integers; the first event uses time zero as its reference.

The same byte stream is compressed with zlib and bz2 at compression level 9 and with lzma at preset 9. The resulting statistic is

$$\ell_c = \frac{8\,|c(S)|}{N}, \tag{14}$$

where S is the serialized event stream, c is the specified compressor, $|c(S)|$ is the compressed size in bytes, N is the number of events, and ℓ_c is the description length in bits per event. We compute the Spearman correlation of each statistic with d_{hold} across the eight datasets used in the capacity analysis. Tied values receive average ranks.

A.3 Comparison with Scale and Compression Statistics

We compare predictive dimension, event count, source-node count, distinct-pair count, and event-stream description lengths by their Spearman correlations with the independently estimated held-out dimension d_{hold}. Statistic-computation details are provided in Appendix A.2.

Figure 7 reports the resulting rank associations. $\widehat{d}_{\text{pred}}$ has the strongest association with d_{hold} ($\rho = 0.892$). The corresponding correlations are 0.546 for source-node count, 0.464 for distinct-pair count, and 0.188 for event count. The zlib, bz2, and lzma description lengths each yield $\rho = 0.082$.

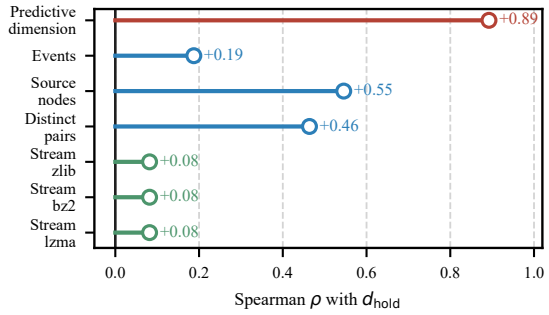


Figure 7: **Competing data statistics.** Spearman correlation of each statistic with d_{hold} across the eight datasets used in the capacity analysis.

B Additional Details for Predictive-Dimension Analysis

B.1 Dependency Interventions

The *time-shuffle* null permutes timestamps and stably resorts the stream; *event-order shuffle* permutes rows directly; *destination shuffle* and *source shuffle* independently permute the respective endpoint column; and *edge-pair shuffle* permutes intact endpoint pairs across event positions. These interventions preserve event count, and pair shuffling additionally preserves the multiset of edges. They are not interchangeable: each removes a different carrier of dependence.

B.2 Compression Controls

The event-stream serialization and compression settings are specified in Appendix A.2. Additional compression controls use a static encoding of sorted unique source–destination pairs and a residual encoding of whether the destination repeats the source's previous destination together with the clipped inter-event time.

C Addition Simplicity Experiments

We report additional experimental results that are not fully included, as shown in Figure 8. The figure compares performance against lightweight baseline methods on four datasets, namely *Wikipedia*, *UCI*, *Enron*, and *Reddit*. The results show that our method achieves state-of-the-art performance on all datasets while maintaining a more concise model architecture than the baselines. Notably,

despite its lower complexity, our method consistently attains higher Average Precision than other more complex baseline methods. Figure 9 provides supplementary results on training-time efficiency and reports the time cost per epoch for our method compared with other baselines. The figure shows that our method not only delivers strong predictive performance but also offers clear advantages in training efficiency, further validating its practicality and scalability.

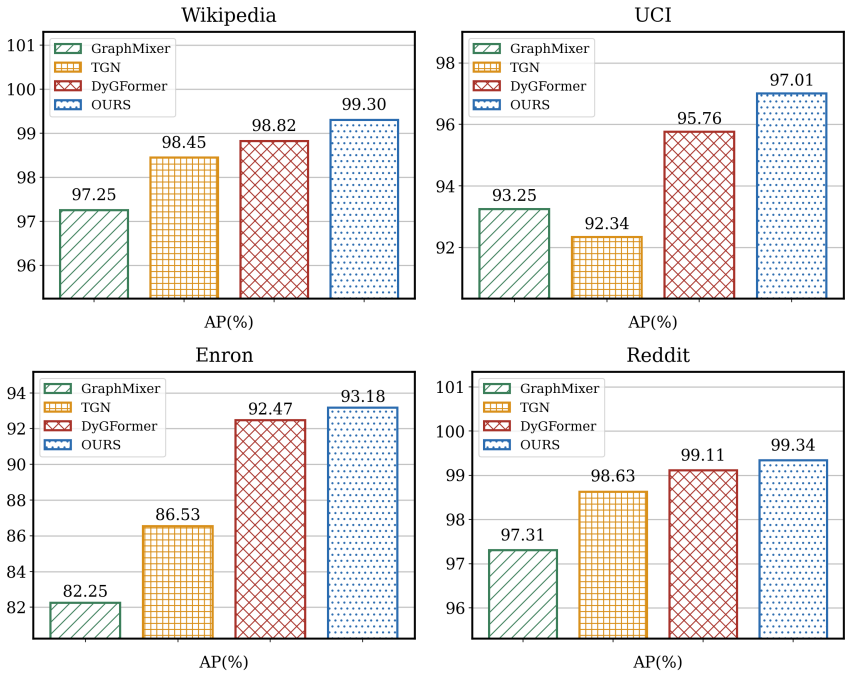


Figure 8: Compared to the performance of lightweight baselines, our method achieves state-of-the-art results on all datasets.

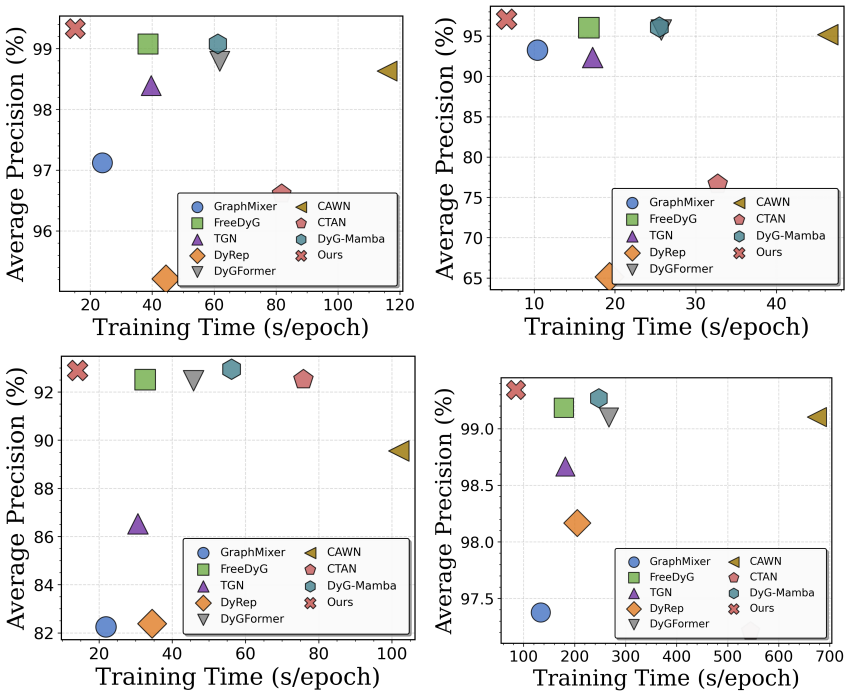


Figure 9: Training time comparison for one epoch across different baselines on multiple datasets: (a) Wikipedia; (b) UCI; (c) Enron; (d) Reddit.

D EXPERIMENTAL DETAILS

Table 6: The statistical information of each component of the dataset used in the experiment.

Dataset	Dim_n	Dim_e	# Nodes	# Edges	Information	Duration	Time Granularity
Wikipedia	-	172	9,227	157,474	Social	1 month	Unix timestamps
Reddit	-	172	10,984	672,447	Social	1 month	Unix timestamps
LastFM	-	-	1,980	1,293,103	Interaction	1 month	Unix timestamps
Enron	-	-	184	125,235	Communication	3 years	Unix timestamps
Social Evo.	-	-	59	90,879	Proximity	77 days	Unix timestamps
Contact	-	-	274	28,244	Proximity	20 days	Unix timestamps
Flights	-	4	13,169	1,470,508	Transportation	3 months	Unix timestamps
UCI	-	-	1,899	59,835	Social	196 days	Unix timestamps

D.1 COMPREHENSIVE DATASET COLLECTION

Our model-side evaluation includes eight conventional temporal graph datasets and four Temporal Graph Benchmark (TGB) datasets.

Wikipedia: This dataset records the collaborative editing activities on the Wikipedia platform. It constructs a bidirectional network where users interact with articles through timestamped editing events. Each interaction includes language features extracted from the edited content as well as temporal user status labels indicating editing permissions. This dataset covers multiple years and reveals the situations in aspects such as knowledge creation, malicious destruction detection, and the evolution of community governance in large-scale collaborative systems.

Reddit: This dataset simulates users' participation in online communities through the submission of posts and interaction with comments. This bidirectional temporal network can capture the dynamic participation of the community, including rich time-varying information about content categories, user activity patterns, and user statuses (including management events). This dataset can be used to analyze the dynamics of social media ecosystems and the formation process of communities.

LastFM: This dataset records the long-term music streaming behavior of users and artists based on their listening events. This two-part temporal network reveals the evolution of preferences, discovery patterns, and consumption habits over time, including metadata such as play counts, music genres, and demographic information, and is highly suitable for research in recommendation systems.

Enron: This dataset, through email communication networks spanning multiple years, reveals the enterprise communication patterns within large organizations. This dynamic graph dataset records the evolution of organizational hierarchical structures, information flow patterns, and communication dynamics during normal operations and crisis periods, thereby enabling the analysis of the network structure of the enterprise.

Social Evo.: This dataset captures the intricate spatio-temporal interaction patterns among college students through proximity detection of mobile devices. The time network presents physical proximity events with high temporal resolution, revealing daily behavioral patterns, the formation of social groups, and the impact of academic schedules on social interactions.

Contact: This dataset is based on the concept of proximity detection, using wearable sensors to monitor the close interactions among student groups. Its accuracy can reach the minute level. The dataset contains detailed metadata about the interaction scenarios and is capable of analyzing social contact patterns, interaction rhythms, and the evolution of community structures.

Flights: This dataset records the evolution of the aviation network during major global events, covering the adjustments of routes and changes in traffic patterns in the international flight network. Its temporal dynamics reflect the operational strategies of airlines, seasonal patterns, and crisis response measures, providing important insights into the stability of the transportation network.

UCI: This dataset records the communication behaviors within the university student population through various digital communication channels. The time graph depicts the interactions among

students and includes metadata about the types of communication and academic backgrounds, thereby revealing how academic pressure and social events affect communication patterns and the formation of interpersonal relationships.

The four TGB datasets are described below Huang et al. (2023).

tgbl-wiki. This dataset records timestamped editing interactions between Wikipedia users and pages over one month. It forms a bipartite graph, with each event representing a user editing a page. The prediction task is to identify the page a user will interact with at a given time.

tgbl-review. This dataset contains timestamped reviews of Amazon electronics products. Users and products form the two node sets, and each interaction represents a user reviewing a product. The prediction task is to identify the product a user will review at a given time.

tgbl-coin. This dataset records cryptocurrency transfers between blockchain addresses. Nodes represent addresses, and directed events represent timestamped transfers. The prediction task is to identify the destination address of a future interaction.

tgbl-comment. This dataset represents a directed Reddit reply network. Nodes are users, and each timestamped event records one user replying to another. The prediction task is to identify the recipient of a user's reply.

D.2 Baseline Algorithm Landscape

DyRepTrivedi et al. (2019): This algorithm establishes the basic method by combining the recurrent neural network architecture and the temporal attention mechanism. It maintains the constantly changing node representations through continuous state updates, which can capture the changes in the local neighborhood as well as the global temporal patterns. By adopting a dual encoder architecture and combining the attention mechanism, it selectively focuses on relevant historical events when updating the node states.

TGNRossi et al. (2020): This algorithm combines a trainable memory library and a message-passing framework to achieve temporal graph learning with enhanced memory capabilities. The architecture sets up dedicated storage modules for each node to store compressed historical information, and uses an attention-based aggregation method to combine the received messages with the stored historical background information.

CAWNWang et al. (2021b): Causal Anonymous Walk-based methods extract meaningful temporal pathways that preserve causality constraints while maintaining structural anonymity. The method generates anonymous walk sequences through time-aware sampling procedures, followed by anonymous encoding that creates generalizable structural representations while preserving essential temporal properties.

GraphMixerCong et al. (2023): This algorithm represents a shift towards a time-encoding encoder based on simplified multi-layer perceptrons. This encoder can achieve comparable performance to similar methods while also reducing computational complexity. GraphMixer emphasizes a fixed time-encoding strategy and employs specialized mixing layers. These mixing layers process temporal and structural information independently through separate paths, and then combine them through a learnable fusion mechanism.

DyGFormerYu et al. (2023): Dynamic Graph Transformer has advanced the development of converter-based temporal graph learning by adopting a specialized architecture with an adjacent co-occurrence encoding scheme and a stitching technique. This method combines temporal position encoding and multi-head attention mechanisms, which have been optimized for temporal graph data and can efficiently handle extended temporal sequences.

FreeDyGTian et al. (2024): Frequency-domain Dynamic Graph learning incorporates Fourier transformation integration to conduct frequency-domain analysis on temporal graph patterns. This method converts time series into frequency-domain representations, where periodic patterns become more prominent. Additionally, it employs learnable frequency-domain operations and a frequency-based message-passing mechanism.

CTANGravina et al. (2024): This algorithm overcomes the limitations of using specially designed mechanisms to capture the complex temporal dependencies between long sequences, which are

applicable for long sequence modeling. This method can handle dependencies at different granularities while maintaining computational feasibility.

DyG-Mamba Li et al. (2025): Dynamic Graph Mamba represents recent developments through state-space model integration, providing efficient mechanisms for processing very long temporal sequences. The method adapts state-space modeling principles with selective state updates based on temporal interactions, achieving linear computational complexity while maintaining competitive performance on temporal graph learning benchmarks.

In addition, we briefly introduce other methods shown in Figure 1, such as EvolveGCN TP-Net and DGM. EvolveGCN evolves GCN parameters over time with an RNN to model dynamic graphs without maintaining explicit node embeddings, enabling adaptation to changing node sets Pareja et al. (2020). TPNet learns time-decayed temporal-walk encodings with efficient random-feature propagation for fast and accurate temporal link prediction Lu et al. (2024). A cross-geometric dynamic graph framework that jointly leverages Euclidean and hyperbolic embeddings with substructure-aware alignment and temporal aggregation to model evolving node and relation dynamics Sun et al..

D.3 Addition Generalization Experiments

We present additional experimental results that are not fully included, as shown in Figure 10. The figure provides a detailed analysis of the generalization performance of PreDyG across multiple datasets and visualizes the training process. By tracking validation performance throughout training, we examine the convergence behavior, generalization capability, and differences from baseline methods. The training curves show that PreDyG achieves faster convergence and a more stable training process on all datasets. The validation loss decreases more rapidly and more smoothly, and PreDyG reaches higher peak performance on both AP and AUC. These results further demonstrate that PreDyG not only delivers strong performance but also offers stable training dynamics and robust generalization.

D.4 Impact of different Negative Sampling Strategies

Tables 7 to 10 present a comprehensive comparison of negative sampling strategies for dynamic link prediction. These four tables cover both transductive and inductive evaluation settings. Under each setting, we test three negative sampling strategies, namely random rnd, historical hist, and inductive ind, and evaluate performance using two complementary metrics, AP and AUC-ROC.

Overall, PreDyG achieves the best performance in most experimental configurations, demonstrating a clear advantage. In the transductive setting shown in Tables 7 and 8, PreDyG is particularly strong under random negative sampling, achieving an average rank of 1.00 and outperforming all baselines including TGN, CAWN, and DyGFormer. On *Wikipedia*, PreDyG reaches 99.30±0.01 AP, which is a clear improvement over the second-best method DyGFormer at 98.82±0.02. Under the more challenging historical negative sampling strategy, performance drops for all methods, yet PreDyG remains competitive with an average rank of 2.38, tying with DyGFormer for the best overall rank. Inductive negative sampling is the most challenging setting, but PreDyG still leads on most datasets, with particularly large gains on *Enron*, *LastFM*, and *Contact*.

The inductive setting shown in Tables 9 and 10 further confirms the generalization ability and robustness of PreDyG. Under this stricter evaluation protocol, the method continues to deliver strong results and achieves the best performance on almost all datasets under random negative sampling. For example, on *UCI*, the AP of PreDyG decreases slightly from 97.01±0.03 in the transductive setting to 95.75±0.09 in the inductive setting, while still remaining clearly better than competing methods.

Notably, datasets differ substantially in their sensitivity to negative sampling strategies, reflecting the complexity and diversity of real-world dynamic graphs, yet PreDyG remains consistently strong. *Social Evolution* exhibits a relatively stable performance pattern with small gaps among methods, and PreDyG maintains a lead. In contrast, *LastFM* shows larger performance disparities, where the advantage of PreDyG is more pronounced. This consistency across datasets indicates that PreDyG is adaptive and robust, and can effectively handle dynamic graph structures with diverse characteristics.

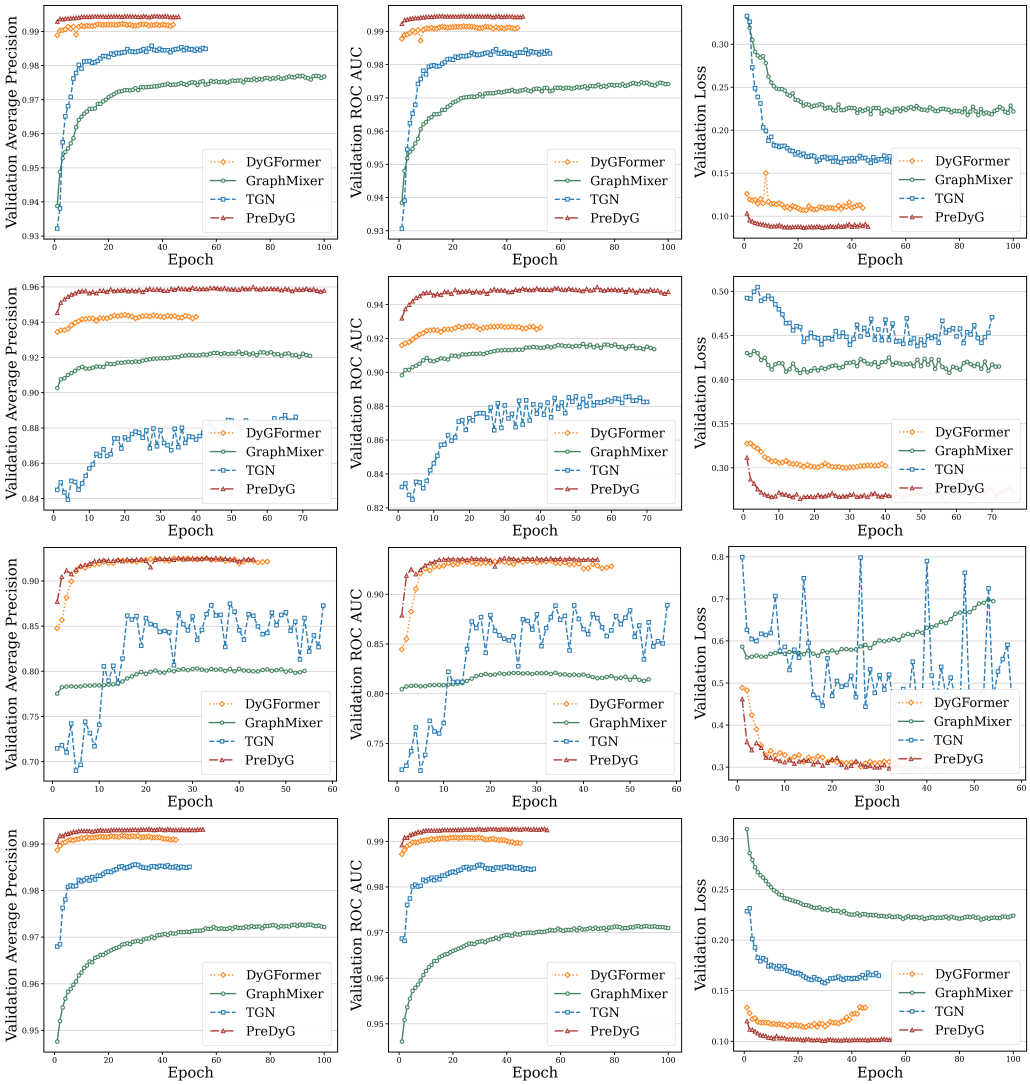


Figure 10: The AP , AUC and loss metrics and generalization gap of multiple training sets with the early stopping mechanism were compared. Training curves comparison across different datasets and evaluation metrics. The first row shows Wikipedia, the second row shows UCI, the third row shows Enron, and the fourth row shows Reddit datasets. From left to right: the average precision (AP) curve, the ROC-AUC curve, and the validation loss curve during training. Each plot displays the PreDyG and other baselines over training time.

Table 7: Average Precision (AP) comparison across different negative sampling strategies (rnd: random, hist: historical, ind: inductive) in transductive link prediction setting.

NSS	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	EdgeBank	TCL	GraphMixer	DyGFormer	PhyDyG
rnd	Wikipedia	96.50 ± 0.18	94.86 ± 0.04	96.94 ± 0.06	98.45 ± 0.05	98.76 ± 0.03	90.37 ± 0.00	96.47 ± 0.14	97.25 ± 0.03	98.82 ± 0.02	**99.30± 0.01**
	Reddit	98.31 ± 0.12	98.22 ± 0.07	98.52 ± 0.02	98.63 ± 0.05	99.11 ± 0.01	94.86 ± 0.00	97.53 ± 0.02	97.31 ± 0.01	99.11 ± 0.01	**99.34 ± 0.01**
	Enron	84.77 ± 0.28	82.38 ± 3.32	71.12 ± 0.96	86.53 ± 1.14	89.56 ± 0.13	83.53 ± 0.00	79.70 ± 0.72	82.25 ± 0.16	92.47 ± 0.15	**93.18± 0.19**
	UCI	89.43 ± 1.06	65.14 ± 2.33	79.63 ± 0.71	92.34 ± 1.02	95.18 ± 0.05	76.20 ± 0.00	89.57 ± 1.61	93.25 ± 0.54	95.76 ± 0.16	**97.01± 0.03**
	LastFM	70.85 ± 2.11	71.92 ± 2.20	73.42 ± 0.18	77.07 ± 3.99	86.99 ± 0.07	79.29 ± 0.00	67.27 ± 2.13	75.61 ± 0.22	93.00 ± 0.10	**93.97± 0.06**
	Social Evo.	89.89 ± 0.52	88.87 ± 0.31	93.16 ± 0.20	93.57 ± 0.21	84.96 ± 0.09	74.95 ± 0.00	93.13 ± 0.17	93.37 ± 0.07	94.47 ± 0.01	**94.68 ± 0.01**
	Flights	95.60 ± 1.72	95.29 ± 0.69	94.03 ± 0.15	97.95 ± 0.16	98.51 ± 0.01	89.35 ± 0.00	91.23 ± 0.02	90.99 ± 0.08	98.25 ± 0.01	**98.79± 0.01**
	Contact	95.31 ± 1.36	95.98 ± 0.15	96.28 ± 0.06	96.89 ± 0.54	90.26 ± 0.30	92.58 ± 0.00	92.44 ± 0.10	91.92 ± 0.03	98.29 ± 0.01	**98.32± 0.01**
	Avg. Rank	6.50	7.50	6.50	4.00	4.38	8.25	7.88	6.75	2.25	**1.00**
hist	Wikipedia	83.01 ± 0.69	79.93 ± 0.60	87.38 ± 0.19	86.86 ± 0.34	71.21 ± 1.66	73.35 ± 0.00	89.05 ± 0.37	**90.90 ± 0.13**	82.23 ± 2.56	85.36± 1.17
	Reddit	80.03 ± 0.31	79.83 ± 0.36	79.55 ± 0.14	81.22 ± 0.59	80.82 ± 0.43	73.59 ± 0.00	77.14 ± 0.18	78.44 ± 0.25	81.57 ± 0.70	**82.28± 0.47**
	LastFM	74.35 ± 3.82	74.92 ± 2.48	71.59 ± 0.30	76.87 ± 4.60	69.86 ± 0.47	73.03 ± 0.00	59.30 ± 2.34	72.47 ± 0.49	81.57 ± 0.48	**83.02± 0.49**
	Enron	69.85 ± 2.70	71.19 ± 2.78	64.07 ± 1.04	73.91 ± 1.81	64.73 ± 0.40	76.53 ± 0.00	70.66 ± 0.33	**77.98 ± 0.86**	75.63 ± 0.76	75.95± 0.39
	Social Evo.	87.44 ± 6.81	93.29 ± 0.42	95.01 ± 0.50	94.45 ± 0.61	85.53 ± 0.41	80.57 ± 0.00	94.74 ± 0.26	94.93 ± 0.25	**96.34 ± 0.17**	96.87± 0.17
	UCI	75.24 ± 5.80	55.10 ± 3.11	68.27 ± 1.41	80.43 ± 2.09	65.30 ± 0.42	65.50 ± 0.00	80.25 ± 2.75	84.11 ± 1.37	82.17 ± 0.84	**85.57± 0.49**
	Flights	66.48 ± 2.52	67.61 ± 1.01	**72.38 ± 0.12**	66.70 ± 1.58	64.72 ± 1.03	70.53 ± 0.00	70.68 ± 0.18	71.47 ± 0.27	66.59 ± 0.53	66.92± 0.54
	Contact	95.31 ± 2.18	96.39 ± 0.19	96.05 ± 0.51	93.05 ± 2.37	84.16 ± 0.52	88.81 ± 0.00	93.86 ± 0.18	93.36 ± 0.41	97.57 ± 0.08	**97.92± 0.01**
	Avg. Rank	6.50	6.12	5.38	5.00	8.75	7.25	5.88	4.00	3.75	**2.38**
ind	Wikipedia	75.65 ± 0.78	70.21 ± 1.61	87.00 ± 0.16	85.62 ± 0.46	74.06 ± 2.59	80.63 ± 0.00	86.76 ± 0.70	**88.59 ± 0.18**	78.29 ± 5.36	80.67± 3.15
	Reddit	86.98 ± 0.15	86.30 ± 0.23	89.59 ± 0.21	88.10 ± 0.22	**91.67 ± 0.26**	85.48 ± 0.00	87.45 ± 0.29	85.26 ± 0.12	89.12 ± 0.42	90.18± 0.18
	LastFM	62.67 ± 4.50	64.41 ± 2.73	71.13 ± 0.14	65.95 ± 6.01	67.48 ± 0.78	75.49 ± 0.00	58.21 ± 0.88	68.12 ± 0.35	73.97 ± 0.46	**77.73± 0.64**
	Enron	68.96 ± 0.98	67.79 ± 1.53	63.94 ± 1.34	70.89 ± 2.70	75.15 ± 0.61	73.89 ± 0.00	71.29 ± 0.30	75.01 ± 0.76	77.41 ± 0.92	**81.74± 0.49**
	Social Evo.	89.82 ± 4.12	93.28 ± 0.46	94.84 ± 0.41	95.13 ± 0.60	88.32 ± 0.29	83.69 ± 0.00	94.90 ± 0.36	94.72 ± 0.32	95.59 ± 0.10	**95.72± 0.12**
	UCI	65.99 ± 1.40	54.79 ± 1.73	68.67 ± 0.86	70.94 ± 0.74	64.61 ± 0.46	57.43 ± 0.00	76.01 ± 1.07	80.10 ± 0.51	72.25 ± 1.68	**83.52± 0.56**
	Flights	69.07 ± 4.03	70.57 ± 1.81	75.48 ± 0.23	71.09 ± 2.75	69.18 ± 1.54	**81.08 ± 0.00**	74.62 ± 0.16	74.87 ± 0.24	70.92 ± 1.74	70.75± 0.98
	Contact	93.43 ± 1.74	94.18 ± 0.12	94.35 ± 0.47	90.18 ± 3.26	89.31 ± 0.25	85.20 ± 0.00	91.35 ± 0.19	90.87 ± 0.32	94.75 ± 0.31	**95.85± 0.23**
	Avg. Rank	7.75	8.00	4.38	5.50	6.75	6.50	5.25	4.75	3.75	**2.38**

Table 8: AUC-ROC comparison across different negative sampling strategies in transductive link prediction setting.

NSS	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	EdgeBank	TCL	GraphMixer	DyGFormer	PhyDyG
rnd	Wikipedia	96.33 ± 0.05	94.37 ± 0.06	96.67 ± 0.07	98.37 ± 0.07	98.54 ± 0.04	90.78 ± 0.00	95.84 ± 0.15	96.92 ± 0.03	98.91 ± 0.02	**99.30± 0.01**
	Reddit	98.31 ± 0.06	98.17 ± 0.02	98.47 ± 0.02	98.60 ± 0.07	99.01 ± 0.01	95.37 ± 0.00	97.42 ± 0.02	97.17 ± 0.02	99.15 ± 0.01	**99.31± 0.01**
	LastFM	70.49 ± 1.70	71.16 ± 1.86	71.59 ± 0.17	78.47 ± 2.94	85.92 ± 0.09	83.77 ± 0.00	64.06 ± 1.17	73.53 ± 0.09	93.05 ± 0.10	**94.10± 0.07**
	Enron	87.96 ± 0.51	84.89 ± 3.04	68.89 ± 1.08	88.32 ± 0.95	90.45 ± 0.17	87.05 ± 0.00	75.74 ± 0.73	84.38 ± 0.24	93.33 ± 0.14	**93.88± 0.02**
	Social Evo.	92.05 ± 0.47	90.76 ± 0.22	94.76 ± 0.16	95.39 ± 0.13	87.34 ± 0.10	81.60 ± 0.00	94.84 ± 0.20	95.23 ± 0.04	96.30 ± 0.01	**96.40± 0.01**
	UCI	90.44 ± 0.45	68.77 ± 2.38	78.53 ± 0.71	92.03 ± 1.11	93.87 ± 0.11	77.30 ± 0.00	87.82 ± 1.39	91.81 ± 0.69	94.49 ± 0.29	**96.22± 0.03**
	Flights	96.21 ± 1.42	95.95 ± 0.60	94.13 ± 0.16	98.22 ± 0.10	98.45 ± 0.01	90.23 ± 0.00	91.21 ± 0.02	91.13 ± 0.01	**98.53 ± 0.01**	98.86± 0.01
	Contact	96.66 ± 0.87	96.48 ± 0.14	96.95 ± 0.05	97.54 ± 0.37	89.99 ± 0.36	94.34 ± 0.00	94.15 ± 0.05	93.94 ± 0.02	98.53 ± 0.01	**98.60± 0.01**
	Avg. Rank	6.25	7.62	6.62	3.88	4.62	8.25	7.88	6.88	2.00	**1.00**
hist	Wikipedia	80.77 ± 0.75	77.74 ± 0.32	82.87 ± 0.26	82.74 ± 0.31	67.84 ± 0.63	77.27 ± 0.00	85.76 ± 0.43	**87.68 ± 0.18**	78.80 ± 1.94	83.37± 0.52
	Reddit	80.52 ± 0.35	80.15 ± 0.21	79.33 ± 0.18	81.11 ± 0.22	80.27 ± 0.28	78.58 ± 0.00	76.49 ± 0.15	77.80 ± 0.10	80.54 ± 0.27	**81.79± 0.22**
	LastFM	75.22 ± 2.33	74.65 ± 2.02	64.27 ± 0.23	77.97 ± 3.00	67.88 ± 0.22	78.09 ± 0.00	47.24 ± 3.12	64.21 ± 0.74	78.78 ± 0.33	**79.68± 0.29**
	Enron	75.39 ± 2.39	74.69 ± 3.55	61.85 ± 1.45	77.09 ± 2.22	65.10 ± 0.34	**79.59 ± 0.00**	67.95 ± 0.92	75.27 ± 1.11	76.55 ± 0.51	74.65± 0.30
	Social Evo.	90.06 ± 3.14	93.12 ± 0.36	93.08 ± 0.58	94.71 ± 0.49	87.43 ± 0.17	85.81 ± 0.00	93.44 ± 0.70	94.39± 0.34	96.44± 0.10	**97.00± 0.15**
	UCI	78.64 ± 3.52	57.91 ± 3.10	58.89 ± 1.60	77.25 ± 2.71	57.86 ± 0.13	69.56 ± 0.00	72.25 ± 3.42	77.54 ± 2.04	76.97 ± 0.22	**79.52± 0.29**
	Flights	68.97 ± 1.88	69.43 ± 0.87	72.20 ± 0.15	68.39 ± 0.96	66.11 ± 0.75	**74.64 ± 0.00**	70.57 ± 0.19	70.37 ± 0.21	68.09 ± 0.44	69.73± 0.32
	Contact	96.35 ± 0.89	96.00 ± 0.23	95.39 ± 0.41	93.76 ± 1.27	83.06 ± 0.33	92.17 ± 0.00	93.34 ± 0.20	93.14 ± 0.32	97.17 ± 0.08	**97.41± 0.07**
	Avg. Rank	4.88	6.38	6.38	4.25	8.75	6.00	6.38	5.38	4.12	**2.50**
ind	Wikipedia	70.96 ± 0.75	67.36 ± 0.94	81.93 ± 0.25	80.97 ± 0.33	70.97 ± 0.97	81.73 ± 0.00	82.19 ± 0.47	**84.28 ± 0.30**	75.09 ± 3.68	81.64± 1.91
	Reddit	83.51 ± 0.12	82.90 ± 0.28	87.13 ± 0.18	84.56 ± 0.26	**88.04 ± 0.32**	85.93 ± 0.00	84.67 ± 0.26	82.21 ± 0.12	86.23 ± 0.53	87.18± 0.57
	LastFM	61.32 ± 3.51	62.15 ± 2.14	63.99 ± 0.18	65.46 ± 4.24	67.92 ± 0.40	**77.37 ± 0.00**	46.93 ± 2.56	60.22 ± 0.33	69.25 ± 0.40	70.36± 0.62
	Enron	70.92 ± 1.06	68.73 ± 1.34	60.45 ± 2.11	71.34 ± 2.50	75.17 ± 0.51	75.00 ± 0.00	67.64 ± 0.87	71.53 ± 0.82	74.07 ± 0.66	**81.06± 0.23**
	Social Evo.	90.01 ± 3.20	93.07 ± 0.38	92.94 ± 0.59	95.24 ± 0.55	89.93 ± 0.12	87.88 ± 0.00	93.44 ± 0.71	94.22 ± 0.31	96.22 ± 0.09	**96.37± 0.03**
	UCI	64.14 ± 1.28	54.25 ± 2.01	60.80 ± 1.03	64.11 ± 1.07	58.06 ± 0.25	58.03 ± 0.00	70.05 ± 1.85	74.59 ± 0.76	65.96 ± 1.21	**75.23± 0.52**
	Flights	69.99 ± 3.13	71.13 ± 1.58	73.47 ± 0.20	71.63 ± 1.71	69.70 ± 0.73	**81.10 ± 0.00**	72.54 ± 0.16	72.21 ± 0.20	69.53 ± 1.17	71.23± 0.82
	Contact	94.47 ± 1.10	94.23 ± 0.15	94.10 ± 0.39	91.64 ± 1.68	87.68 ± 0.23	85.87 ± 0.00	91.23 ± 0.19	90.96 ± 0.25	95.01 ± 0.15	**95.47± 0.09**
	Avg. Rank	7.00	7.62	5.38	5.50	6.25	5.38	5.62	5.38	4.50	**2.38**

Table 9: Average Precision (AP) comparison across different negative sampling strategies in inductive link prediction setting.

NSS	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	TCL	GraphMixer	DyGFormer	PhyDyG
rnd	Wikipedia	94.82 ± 0.23	92.43 ± 0.38	96.22 ± 0.06	97.83 ± 0.07	98.24 ± 0.03	96.22 ± 0.15	96.65 ± 0.02	98.59 ± 0.03	**98.96± 0.01**
	Reddit	96.50 ± 0.16	96.09 ± 0.11	97.09 ± 0.06	97.50 ± 0.09	98.62 ± 0.01	94.09 ± 0.11	95.26 ± 0.02	98.84 ± 0.02	**99.00± 0.01**
	LastFM	81.61 ± 3.81	83.02 ± 1.44	78.63 ± 0.34	81.45 ± 4.29	89.42 ± 0.06	73.53 ± 1.67	82.11 ± 0.40	94.23 ± 0.06	**95.18± 0.02**
	Enron	80.72 ± 1.42	74.55 ± 3.93	67.05 ± 1.54	77.94 ± 1.05	86.35 ± 0.51	76.14 ± 0.82	75.88 ± 0.46	89.76 ± 0.34	**89.90± 0.19**
	Social Evo.	91.96 ± 0.49	90.04 ± 0.50	91.41 ± 0.13	90.77 ± 0.89	79.94 ± 0.20	91.55 ± 0.08	91.86 ± 0.06	93.14 ± 0.06	**93.20± 0.08**
	UCI	79.86 ± 1.46	57.48 ± 1.85	79.54 ± 0.51	88.12 ± 2.05	92.73 ± 0.07	87.36 ± 2.04	91.19 ± 0.41	94.54 ± 0.15	**95.75± 0.09**
	Flights	94.74 ± 0.41	92.88 ± 0.72	88.73 ± 0.33	95.03 ± 0.64	97.06 ± 0.02	83.41 ± 0.10	83.03 ± 0.05	**97.79 ± 0.02**	97.54± 0.01
	Contact	94.34 ± 1.46	92.18 ± 0.42	95.87 ± 0.07	93.82 ± 1.02	89.55 ± 0.27	91.11 ± 0.09	90.59 ± 0.08	98.03 ± 0.02	**98.07± 0.01**
	Avg. Rank	5.38	7.12	6.50	5.12	4.50	7.12	6.25	1.88	**1.12**
hist	Wikipedia	68.69 ± 0.37	62.18 ± 1.24	84.17 ± 0.21	81.76 ± 0.36	67.27 ± 1.66	82.20 ± 2.16	**87.60 ± 0.28**	71.42 ± 4.44	76.41± 3.16
	Reddit	62.34 ± 0.50	61.60 ± 0.73	63.47 ± 0.36	64.85 ± 0.84	63.67 ± 0.42	60.83 ± 0.29	64.50 ± 0.23	65.37 ± 0.59	**67.25± 0.69**
	LastFM	70.39 ± 4.29	71.45 ± 1.76	76.27 ± 0.28	66.65 ± 6.09	71.33 ± 0.47	65.78 ± 0.63	76.42 ± 0.25	76.35 ± 0.52	**77.73± 0.47**
	Enron	65.86 ± 3.73	62.08 ± 2.24	61.40 ± 1.34	62.91 ± 1.16	60.70 ± 0.32	67.11 ± 0.63	72.37 ± 1.40	67.07 ± 0.65	**81.74± 0.39**
	Social Evo.	88.51 ± 0.91	88.72 ± 1.06	93.97 ± 0.50	90.66 ± 1.65	79.83 ± 0.37	94.10 ± 0.33	94.01 ± 0.43	94.97 ± 0.24	**95.73± 0.25**
	UCI	63.11 ± 2.29	52.47 ± 2.05	70.52 ± 0.97	70.80 ± 0.80	64.54 ± 0.45	76.71 ± 0.98	81.66 ± 0.47	72.13 ± 1.90	**83.52± 0.67**
	Flights	61.01 ± 1.66	62.83 ± 1.31	64.72 ± 0.38	59.31 ± 1.45	56.82 ± 0.55	64.50 ± 0.23	**65.28 ± 0.28**	57.11 ± 0.25	57.77± 0.33
	Contact	90.42 ± 2.37	89.22 ± 0.69	94.15 ± 0.43	88.13 ± 1.48	74.20 ± 0.79	90.44 ± 0.16	89.91 ± 0.40	93.56 ± 0.54	**95.20± 0.47**
	Avg. Rank	6.50	7.00	4.38	5.75	7.75	4.62	2.75	4.00	**2.25**
ind	Wikipedia	68.70 ± 0.36	62.19 ± 1.26	84.17 ± 0.19	81.77 ± 0.32	67.24 ± 1.66	82.20 ± 2.21	**87.60 ± 0.25**	71.42 ± 4.45	76.42± 1.15
	Reddit	62.32 ± 0.51	61.58 ± 0.70	63.40 ± 0.40	64.84 ± 0.83	63.65 ± 0.39	60.81 ± 0.22	64.49 ± 0.26	65.35 ± 0.63	**67.30± 0.49**
	LastFM	70.39 ± 4.33	71.45 ± 1.71	76.28 ± 0.21	69.46 ± 4.64	71.33 ± 0.44	65.78 ± 0.64	76.42 ± 0.22	76.35 ± 0.50	**77.73± 0.50**
	Enron	65.86 ± 3.68	62.08 ± 2.25	61.40 ± 1.30	62.90 ± 1.19	60.72 ± 0.37	67.11 ± 0.60	72.37 ± 1.40	67.07 ± 0.62	**81.74± 0.34**
	Social Evo.	88.51 ± 0.87	88.72 ± 1.11	93.97 ± 0.53	90.65 ± 1.58	79.83 ± 0.37	94.10 ± 0.35	94.01 ± 0.45	95.11 ± 0.22	**95.72± 0.28**
	UCI	63.16 ± 2.29	52.47 ± 2.08	70.49 ± 0.95	70.73 ± 0.82	64.54 ± 0.49	76.65 ± 1.02	81.64 ± 0.45	72.13 ± 1.90	**83.52± 0.75**
	Flights	61.01 ± 1.64	62.83 ± 1.30	64.72 ± 0.36	59.32 ± 1.44	56.82 ± 0.60	64.50 ± 0.23	**65.29 ± 0.26**	57.11 ± 0.22	57.81± 0.18
	Contact	90.43 ± 2.35	89.22 ± 0.69	94.14 ± 0.46	88.12 ± 1.46	74.19 ± 0.80	90.43 ± 0.17	89.91 ± 0.39	93.55 ± 0.48	**95.21± 0.44**
	Avg. Rank	6.38	7.00	4.38	5.75	7.75	4.75	2.75	4.00	**2.25**

Table 10: AUC-ROC comparison across different negative sampling strategies in inductive link prediction setting.

NSS	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	TCL	GraphMixer	DyGFormer	PhyDyG
rnd	Wikipedia	94.33 ± 0.28	91.49 ± 0.47	95.90 ± 0.10	97.72 ± 0.03	98.03 ± 0.03	95.57 ± 0.18	96.30 ± 0.06	98.48 ± 0.03	**98.93± 0.01**
	Reddit	96.52 ± 0.10	96.05 ± 0.11	96.98 ± 0.07	97.39 ± 0.08	98.42 ± 0.02	93.80 ± 0.03	94.97 ± 0.02	98.71 ± 0.01	**98.93± 0.01**
	LastFM	81.13 ± 3.37	82.24 ± 1.48	76.99 ± 0.26	82.61 ± 3.15	87.82 ± 0.15	70.84 ± 0.82	80.37 ± 0.21	94.08 ± 0.07	**95.12± 0.02**
	Enron	81.96 ± 1.37	76.34 ± 4.20	64.63 ± 1.72	78.83 ± 1.13	87.02 ± 0.47	72.33 ± 0.96	76.51 ± 0.71	90.69 ± 0.29	**90.84± 0.17**
	Social Evo.	93.70 ± 0.32	91.18 ± 0.52	93.41 ± 0.17	93.43 ± 0.57	84.73 ± 0.30	93.71 ± 0.17	94.09 ± 0.10	95.29 ± 0.03	**95.37± 0.05**
	UCI	78.80 ± 0.97	58.08 ± 1.79	77.64 ± 0.41	86.68 ± 2.26	90.40 ± 0.13	84.49 ± 1.84	89.30 ± 0.59	92.63 ± 0.10	**94.40± 0.13**
	Flights	95.21 ± 0.33	93.56 ± 0.73	88.64 ± 0.32	95.92 ± 0.40	96.86 ± 0.02	82.48 ± 0.01	82.27 ± 0.03	**97.80 ± 0.02**	97.64± 0.03
	Contact	95.37 ± 0.93	91.89 ± 0.41	96.53 ± 0.14	94.84 ± 0.71	89.07 ± 0.30	93.05 ± 0.11	92.83 ± 0.02	98.30 ± 0.02	**98.37± 0.03**
	Avg. Rank	5.62	7.38	6.62	4.62	4.50	7.12	6.12	1.88	**1.12**
hist	Wikipedia	61.86 ± 0.51	57.54 ± 1.12	78.38 ± 0.16	75.75 ± 0.28	62.04 ± 0.63	79.79 ± 0.97	**82.87 ± 0.17**	68.33 ± 2.82	75.23± 2.03
	Reddit	61.69 ± 0.40	60.45 ± 0.40	64.43 ± 0.30	64.55 ± 0.47	64.94 ± 0.20	61.43 ± 0.26	64.27 ± 0.16	64.81 ± 0.24	**67.14± 0.41**
	LastFM	68.44 ± 3.25	68.79 ± 1.08	69.89 ± 0.24	66.99 ± 5.65	67.69 ± 0.23	55.88 ± 1.84	70.07 ± 0.22	**70.73 ± 0.33**	70.36± 0.29
	Enron	65.32 ± 3.56	61.50 ± 2.52	57.84 ± 2.18	62.68 ± 1.06	62.25 ± 0.42	64.06 ± 1.05	68.20 ± 1.62	65.78 ± 0.44	**81.06± 0.48**
	Social Evo.	88.53 ± 0.53	87.93 ± 1.04	91.87 ± 0.69	92.10 ± 1.26	83.54 ± 0.21	93.28 ± 0.59	93.62 ± 0.33	96.12 ± 0.08	**96.37± 0.08**
	UCI	60.24 ± 1.95	51.25 ± 2.35	62.32 ± 1.14	62.69 ± 0.92	56.39 ± 0.08	70.46 ± 1.90	**75.98 ± 0.80**	65.55 ± 1.01	75.23± 0.35
	Flights	60.72 ± 1.31	61.99 ± 1.42	63.38 ± 0.29	59.66 ± 1.03	56.58 ± 0.46	**63.48 ± 0.24**	63.30 ± 0.19	56.05 ± 0.22	61.25± 0.39
	Contact	90.80 ± 1.16	88.88 ± 0.64	93.76 ± 0.40	88.84 ± 1.43	74.79 ± 0.34	90.37 ± 0.19	90.04 ± 0.30	94.14 ± 0.27	**95.48± 0.24**
	Avg. Rank	6.12	7.38	4.75	5.88	7.12	4.62	3.12	3.75	**2.25**
ind	Wikipedia	61.87 ± 0.49	57.54 ± 1.07	78.38 ± 0.22	75.76 ± 0.31	62.02 ± 0.66	79.79 ± 0.94	**82.88 ± 0.23**	68.33 ± 2.84	76.24± 1.43
	Reddit	61.69 ± 0.36	60.44 ± 0.39	64.39 ± 0.23	64.55 ± 0.49	64.91 ± 0.21	61.36 ± 0.27	64.27 ± 0.14	64.80 ± 0.24	**67.18± 0.25**
	LastFM	68.44 ± 3.28	68.79 ± 1.08	69.89 ± 0.29	66.99 ± 5.59	67.68 ± 0.24	55.88 ± 1.88	70.07 ± 0.24	**70.73 ± 0.33**	70.36± 0.36
	Enron	65.32 ± 3.60	61.50 ± 2.49	57.83 ± 2.17	62.68 ± 1.08	62.27 ± 0.39	64.05 ± 1.03	68.19 ± 1.61	65.79 ± 0.38	**81.06± 0.30**
	Social Evo.	88.53 ± 0.54	87.93 ± 1.05	91.88 ± 0.74	92.10 ± 1.25	83.54 ± 0.23	93.28 ± 0.62	93.62 ± 0.35	96.03 ± 0.05	**96.37± 0.05**
	UCI	60.27 ± 1.94	51.26 ± 2.38	62.29 ± 1.21	62.66 ± 0.88	56.39 ± 0.14	70.42 ± 1.96	**75.97 ± 0.81**	65.58 ± 1.03	75.23± 0.29
	Flights	60.72 ± 1.26	61.99 ± 1.41	63.40 ± 0.29	59.66 ± 1.02	56.58 ± 0.41	**63.49 ± 0.22**	63.32 ± 0.16	56.05 ± 0.19	61.23± 0.30
	Contact	90.80 ± 1.19	88.87 ± 0.70	93.76 ± 0.43	88.85 ± 1.37	74.79 ± 0.35	90.37 ± 0.12	90.04 ± 0.27	94.14 ± 0.26	**95.47± 0.23**
	Avg. Rank	6.12	7.38	4.75	6.00	7.12	4.62	3.12	3.75	**2.12**