IDG: INTRINSIC ASSESSMENT AND INFORMATION-THEORETIC LEARNING FOR DYNAMIC GRAPH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Dynamic graph neural networks (DyGNNs) aim to transform evolving graph structures into informative low-dimensional representations to support various downstream tasks, such as link prediction and node classification. However, assessing the quality of latent embeddings remains difficult because existing methods mainly rely on downstream task performance. Such end-to-end assessment lacks interpretable metrics for directly evaluating the latent embeddings themselves. This reliance leads to repeated tuning trials, limited diagnostic insight, and unstable model selection across tasks and datasets. To address this issue, we propose **IDG**, which introduces **I**ntrinsic Assessment and Information-Theoretic Learning for **D**ynamic **G**raph Neural Networks. Specifically, IDG contains two key components: (i) structure mutual information (SMI), which uses spectral entropy to measure structural information shared by inputs and latent embeddings, and (ii) feature mutual information (FMI), which uses f-divergence to capture the dependence between latent embeddings and node attribute dynamics. To further demonstrate the practical utility of these metrics, we integrate them into a representation learning objective to guide embedding optimization. Comprehensive experiments show positive empirical associations between the proposed metrics and downstream performance across diverse benchmarks and model architectures, providing interpretable clues about model efficacy, failure modes, and robustness patterns across settings under the evaluated protocols. The code is available at `https://anonymous.4open.science/r/IDG_ICLR-924F`.

1 INTRODUCTION

Dynamic graphs are fundamental for modeling evolving real-world networks and are widely employed in domains such as community detection (Sattar et al., 2025) and temporal recommendation (Kumar et al., 2019; Jiao et al., 2025), where entities often interact through complex and time-varying relations. As a prominent paradigm, dynamic graph neural networks (DyGNNs) are important because they combine graph representation learning with temporal modeling to simultaneously capture the non-Euclidean topological structure and the temporal evolution characteristics of dynamic graphs (Feng et al., 2026). Recent intrinsic diagnostics characterize representation diversity across scales (Limbeck et al., 2024). Beyond downstream scores, we ask how much structural and feature information a latent embedding preserves across successive snapshots.

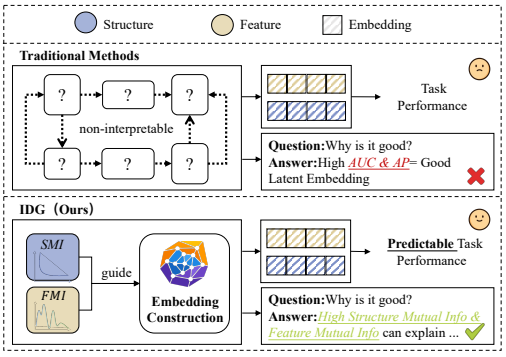

Figure 1: Traditional assessment relies on downstream scores, whereas IDG directly assesses latent embeddings with SMI and FMI.

Existing studies related to dynamic graph latent embedding quality can be broadly organized into two lines. **(i) end-to-end DyGNN learning**, which focuses on designing temporal graph encoders and assesses latent embeddings mainly through downstream tasks. Recent examples include causal

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.

invariant learning (Zhang et al., 2026) and adaptive memory retention (De Castelli et al., 2026). Typically, DySAT (Sankar et al., 2020) stacks structural self-attention and temporal self-attention to learn node latent embeddings from graph snapshots. Its effectiveness is evaluated through downstream prediction performance. **(ii) intrinsic representation learning and assessment** (Wang & Isola, 2020; Veličković et al., 2019; Garrido et al., 2023), which studies the intrinsic properties of latent embeddings. Typically, DGI (Veličković et al., 2019) maximizes mutual information between local patch representations and a global graph summary to encourage latent embeddings that retain graph-level information. DGI supplies a self-supervised training objective, while RankMe and smoothness measures supply post hoc diagnostics. These approaches provide direct views of representation properties, but existing metrics are mostly developed for static graphs or general representation learning rather than dynamic graph latent embeddings with coupled structural and feature evolution. As a result, current DyGNN research still lacks a unified way to connect intrinsic latent embedding quality with downstream performance.

This gap leads to two basic limitations. **(i) lack of intrinsic assessment for dynamic latent embeddings.** Existing DyGNN methods predominantly assess latent embeddings through downstream performance rather than direct criteria on the latent embeddings themselves. Because assessment protocols vary across tasks and datasets, this task-dependent reliance can produce unstable or inconsistent conclusions about latent embedding quality. **(ii) insufficient interpretability of latent embedding quality.** Existing latent embedding assessment metrics provide useful direct signals in static or general representation settings, but they do not explicitly characterize how dynamic graph latent embeddings preserve evolving topology and node features together. This makes it difficult to explain why a given latent embedding supports effective prediction or to diagnose the failure modes of suboptimal latent embeddings. Ultimately, the absence of intrinsic assessment leads to optimization by repeated trials and limits principled model design.

In this paper, we propose IDG, an information-theoretic framework that first establishes intrinsic assessment metrics for DyGNNs and then develops a learning method from these metrics. Specifically, it addresses two fundamental questions: **(i)** *How can we design a suite of interpretable metrics to directly assess the quality of latent embeddings?* **(ii)** *How can we construct a learning framework based on these metrics to ensure the generation of effective latent embeddings?* To achieve this, we treat structure and node features as complementary signals in dynamic graph data and introduce structure mutual information (SMI) and feature mutual information (FMI) to quantify latent embedding quality from structural and feature perspectives, respectively. Based on this, IDG derives an information-theoretic bound among the data, tasks, and latent embedding, and then optimizes the latent embedding under this constraint. Through correlation analysis with vanilla DyGNN models, we reveal clear links between the proposed metrics and downstream task performance, yielding interpretability and theoretical insight for future dynamic graph modeling. Our core contributions are:

- We design SMI based on spectral entropy and FMI based on f-divergence, providing interpretable criteria for dynamic graph latent embeddings across evolving graphs and tasks.
- We establish IDG, a DyGNN framework with a bottleneck objective, that preserves the evolution of structure and features while remaining tractable through a local variational approximation of dynamic graph dependencies.
- We further include correlation analysis, ablations, cross-task and cross-dataset sensitivity analysis, post hoc perturbation studies, and large-scale efficiency comparisons, including recent snapshot-based baselines for broader empirical validation.

2 PRELIMINARIES

Dynamic graphs A snapshot sequence is $\mathcal{G} = \{G_t\}_{t=1}^{T}$, where $G_t = (V_t, E_t, X_t, A_t)$ contains vertices, edges, features $X_t \in \mathbb{R}^{|V_t| \times d_x}$, and adjacency $A_t \in \mathbb{R}^{|V_t| \times |V_t|}$; d_x is the attribute dimension. An event stream $\mathcal{E} = \{(u_i, v_i, t_i, \mathbf{e}_i)\}_{i=1}^{N}$ records N interactions with endpoints u_i, v_i, timestamp t_i, and event attributes $\mathbf{e}_i$. IDG processes snapshots, obtained by chronological discretization when the source is an event stream. Timestamp windows are left-closed and right-open (Appendix G.1).

Mutual information Mutual information (Cover & Thomas, 2006) is an information-theoretic measure that quantifies dependence between random variables, capturing linear and nonlinear cor-

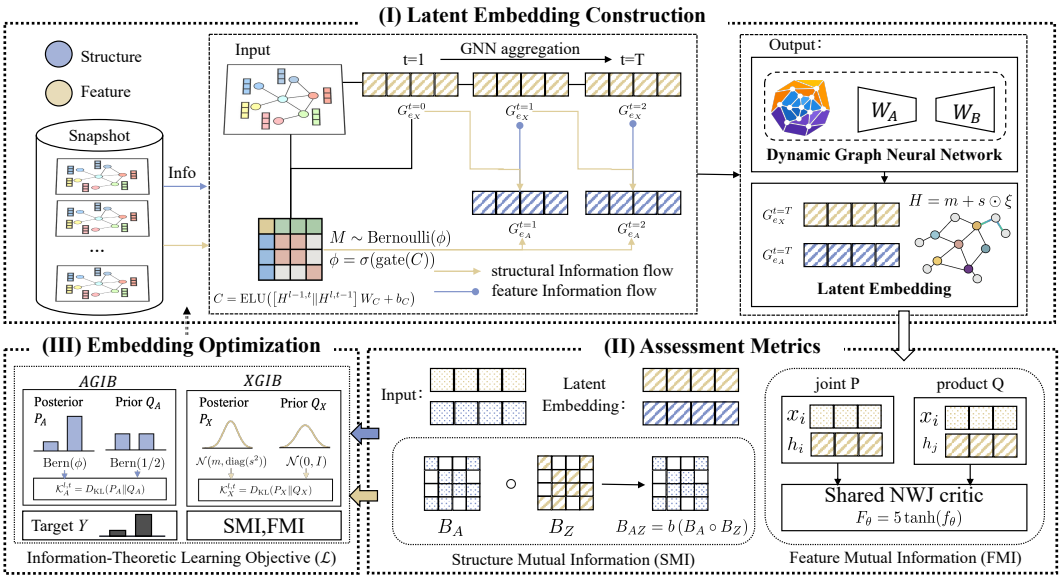

Figure 2: IDG. (I) Causal snapshot updates construct structural gates and feature states. (II) Paired input and latent embedding records yield SMI and FMI. (III) Posterior-to-prior KL terms, task prediction, and SMI/FMI form the information-theoretic learning objective; the return path denotes optimization.

relations without distributional assumptions. For discrete random variables X and Y with finite entropy, joint distribution $\mathbb{P}_{XY}$ and marginal distributions $\mathbb{P}_X$, $\mathbb{P}_Y$, it is defined as $I(X;Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X) = D_{\mathrm{KL}}(\mathbb{P}_{XY} \parallel \mathbb{P}_X \otimes \mathbb{P}_Y)$, where $H(\cdot)$ denotes Shannon entropy, $H(\cdot \mid \cdot)$ conditional entropy, and $D_{\mathrm{KL}}(\cdot \parallel \cdot)$ Kullback-Leibler divergence. For continuous variables, the same KL definition uses the joint law and the product of its marginals, with finite MI when the log density ratio is integrable. Mutual information is pivotal in graph learning, but exact computation is intractable for high-dimensional continuous data, with traditional estimators lacking scalability. The Nguyen-Wainwright-Jordan (NWJ) variational representation (Nguyen et al., 2010; Poole et al., 2019) expresses KL divergence through a joint expectation and an independent-marginal expectation. A neural critic makes this lower bound differentiable and computable from sampled attribute and latent embedding pairs, as used for FMI below.

Information bottleneck The information bottleneck (IB) principle (Tishby et al., 2000) provides an information-theoretic framework for learning robust latent embeddings Z. This objective is formally defined by minimizing the following bottleneck Lagrangian:

$$\mathcal{L}_{\mathrm{IB}} = -I(C;Z) + \beta I(D;Z), \tag{1}$$

where D is the input, C is the target, and $\beta > 0$ controls compression relative to prediction. We treat Z as the bottleneck representation and employ variational bounds to preserve task-relevant information while compressing the input.

3 The Framework of IDG

We construct IDG around the three modules shown in Figure 2. **(I)** latent embedding construction maps raw dynamic graph data into a latent embedding by propagating structural and feature information through local causal updates. **(II)** assessment metrics quantify the latent embedding through structure mutual information (SMI) and feature mutual information (FMI), measuring input and latent embedding structural and feature dependence. **(III)** embedding optimization introduces the information-theoretic learning objective that removes redundant information while retaining task-relevant signals.

3.1 LATENT EMBEDDING CONSTRUCTION

The latent embedding is represented by $G_e = (V_{G_e}, E_{G_e}, M, H)$, with latent vertex/edge sets V_{G_e}, E_{G_e}, where M is a binary latent adjacency matrix and H is the continuous feature state. We use G for the observed dynamic graph sequence, G' for the task target (link, node, or graph labels), and G_e for the final latent embedding. The symbols $G_{e_A}^{(t)}$ and $G_{e_X}^{(t)}$ denote, respectively, its discrete structural state and continuous feature state at time t. Given $G = \{G_1, \ldots, G_T\}$, we use a common padded vertex index set $\bar{V}$ when comparing snapshots with different node sets. Let $\mathcal{S} = \{S^i\}_{i=1}^T$ collect the node alignments. The assignment matrix S^i satisfies the one-hot constraint $\sum_\alpha S^i_{a\alpha} = 1$ for every observed node a. The graph-prototype likelihood (Han et al., 2015) provides a generative interpretation of this structured state. Its reconstruction score is:

$$\mathcal{W}(G; G_e, \mathcal{S}) = \prod_{i,a,\alpha} \left[B^i_{a\alpha} P_e^{|V^i|} \exp\left(\log \frac{1-P_e}{P_e} \sum_{b \in V^i} c^i_{ab\alpha} \right) \right]^{S^i_{a\alpha}}. \tag{2}$$

Here i indexes snapshots, a, b observed nodes, and α latent nodes; $c^i_{ab\alpha}$ indicates agreement of observed and identity-aligned latent edges. The normalized assignment prior is $B^i_{a\alpha}$, and $0 < P_e < 1/2$ is the edge-error probability. Appendix C gives the full agreement formula and derivation. This score provides the generative interpretation; the causal encoder uses persistent identities and minimizes equation 10.

3.2 ASSESSMENT METRICS

The input comprises structure and features, $D = (A, X)$. Module (II) assesses their dependence with paired node/window latent embeddings from any encoder. Each baseline supplies its latent embedding at inference; IDG supplies its posterior mean. SMI and FMI thus share an input and latent embedding interface across all models (Appendix A.7).

3.2.1 STRUCTURE MUTUAL INFORMATION (SMI)

SMI is the mutual information between the input structure and its latent embedding in a spectral kernel representation. We sample paired node/window records $(a_i, z_i^l)_{i=1}^b$: a_i describes the observed neighborhood of node v_i at time t_i, and z_i^l is its layer-l latent embedding, using the posterior mean for IDG. This pairing measures structural fidelity through the dependence retained between observed neighborhoods and latent embeddings. Appendix D.1 specifies the sparse structural descriptors and establishes the associated information identities.

For Gaussian kernels k_A, k_Z with fixed positive bandwidths, define trace-one Gram matrices B_A and B_Z^l. Their joint matrix uses the Hadamard product $\circ$, whose entries are multiplied componentwise. Spectral entropy is the von Neumann entropy of a positive semidefinite trace-one matrix (Sanchez Giraldo et al., 2015; Watrous, 2018):

$$\begin{aligned} (B_A)_{ij} &= b^{-1} k_A(a_i, a_j), \quad (B_Z^l)_{ij} = b^{-1} k_Z(z_i^l, z_j^l), \\ B_{AZ}^l &= b(B_A \circ B_Z^l), \qquad S(B) = -\operatorname{Tr}(B \log B) = -\sum_j \nu_j \log \nu_j. \end{aligned} \tag{3}$$

Here $k_A(a,a) = k_Z(z,z) = 1$, ν_j are the eigenvalues of B, $0 \log 0 = 0$, and natural logarithms give scores in nats. The product kernel couples each input record to its own latent embedding, yielding

$$\mathrm{SMI}(G; G_e) = \sum_{l=1}^{L} I_{\mathrm{sp}}^l, \qquad I_{\mathrm{sp}}^l = S(B_A) + S(B_Z^l) - S(B_{AZ}^l). \tag{4}$$

The product kernel represents a mixture of paired input and latent feature states. Its partial traces give the two marginal states, whose nonzero spectra match the corresponding Gram matrices. Their quantum relative entropy therefore equals I_{sp}^l, with $0 \le I_{\mathrm{sp}}^l \le \min\{S(B_A), S(B_Z^l)\} \le \log b$ (Appendix D.1). A stable graph can have positive SMI through variation across nodes; constant descriptors or collapsed latent embeddings give zero. Common node relabeling and orthogonal changes of latent coordinates preserve the score.

SMI measures dependence at the resolution of the structural descriptor and kernel bandwidth. We use the same paired-record construction for IDG training and for final-layer assessment of every frozen model, including IDG. Local kernel preparation links structure mutual information to recoverable input information (Appendix E.3). Section 4.3 evaluates its positive association with downstream performance under shared sampling and normalization.

3.2.2 FEATURE MUTUAL INFORMATION (FMI)

Let X_{obs} be an observed attribute record and H_e its paired latent embedding from any fixed encoder. Define $\mathrm{FMI}(G; G_e) := I(X_{\mathrm{obs}}; H_e)$, with $P = \mathbb{P}_{X_{\mathrm{obs}} H_e}$ and $Q = \mathbb{P}_{X_{\mathrm{obs}}} \otimes \mathbb{P}_{H_e}$. Here G_e denotes the model's latent embedding; deterministic latent embeddings induce a valid finite empirical record law. For $P \ll Q$, Fenchel's inequality $r \log r \geq rf - e^{f-1}$, integrated with $r = dP/dQ$, proves the NWJ lower bound (Nguyen et al., 2010; Poole et al., 2019):

$$\mathrm{FMI}(G; G_e) \geq \sup_{F \in \mathcal{F}_\Theta} \left\{ \mathbb{E}_P[F] - \mathbb{E}_Q[\exp(F-1)] \right\}, \tag{5}$$

Here $P \ll Q$ means absolute continuity, dP/dQ is the density ratio, and the population KL and critic expectations are assumed finite. $\mathcal{F}_\Theta$ is the critic class; $F^* = 1 + \log(dP/dQ)$ attains equality if finite and admitted (otherwise a limiting supremum may be needed). For n joint pairs $(x_i, h_i) \sim P$ and independent pairs $(\widetilde{x}_i, \widetilde{h}_i) \sim Q$, estimate

$$\widehat{\mathrm{FMI}}_n = \sup_{\theta \in \Theta} \left[\frac{1}{n} \sum_{i=1}^{n} F_\theta(x_i, h_i) - \frac{1}{n} \sum_{i=1}^{n} \exp\Big(F_\theta(\widetilde{x}_i, \widetilde{h}_i) - 1 \Big) \right]. \tag{6}$$

If the empirical objective $\widehat{J}$ satisfies $\sup_\theta |\widehat{J}_\theta - J_\theta| \leq \varepsilon_{\mathrm{est}}$ and optimization is within $\varepsilon_{\mathrm{opt}}$ of its supremum, then $I - J_{\widehat{\theta}} \leq \varepsilon_{\mathrm{app}} + 2\varepsilon_{\mathrm{est}} + \varepsilon_{\mathrm{opt}}$, where $\varepsilon_{\mathrm{app}} = I - \sup_\theta J_\theta$. This follows by inserting the two suprema and using the uniform deviation twice. Dependence assumptions, held-out assessment, and the distinction between empirical scores and population lower bounds are detailed in Appendices D.2 and A.7.

3.3 EMBEDDING OPTIMIZATION

Module (III) of Figure 2 balances retained structural and feature information with compression of task-irrelevant variation through the information-theoretic learning objective.

Following the graph information bottleneck formulation (Wu et al., 2020), we use the bottleneck objective

$$\mathcal{L}_{\mathrm{GIB}}(G_e) = -I(G'; G_e) + \beta I(G; G_e). \tag{7}$$

Here $\beta > 0$ controls compression. We impose encoder locality: the update for node v uses its past state, current features, and at most k_{nbr} neighbors within look-back horizon Δt. This locality specifies the encoder architecture and permits arbitrary dependence in the observed graph distribution. All temporal readouts use only the observed prefix $G^{1:t}$, never future snapshots or labels.

At layer l and time t, sample structure before features, conditioning on the observed prefix $G^{1:t}$ and preceding random states $H_A^{l,t}$ or $H_X^{l,t}$. These histories include earlier layers and windows; $H_X^{l,t}$ also includes the current structural draw. Prefix conditioning covers the causal neighbor cache. Appendix E.1 gives the factorization and chain rule. The resulting surrogate is:

$$\mathcal{U}_{\mathrm{GIB}} := \beta \sum_{t \in S_A} \mathrm{AGIB}^{(t)} + \beta \sum_{t \in S_X} \mathrm{XGIB}^{(t)} - \widehat{I}_{\mathrm{pred}}(G'; G_e). \tag{8}$$

Here S_X and S_A denote the index sets of snapshots used for feature-state and structural-state updates, respectively. The terms $\mathrm{XGIB}^{(t)}$ and $\mathrm{AGIB}^{(t)}$ are defined as:

$$\mathrm{AGIB}^{(t)} = \sum_l \mathbb{E} D_{\mathrm{KL}} \Big(P(G_{e_A}^{l,t} \mid G^{1:t}, H_A^{l,t}) \| Q_A^{l,t} \Big),$$

$$\mathrm{XGIB}^{(t)} = \sum_l \mathbb{E} D_{\mathrm{KL}} \Big(P(G_{e_X}^{l,t} \mid G^{1:t}, H_X^{l,t}) \| Q_X^{l,t} \Big).$$

For each retained stochastic update Z_j with history $\mathcal{H}_j$ and a fixed full-support prior Q_j, adding and subtracting $\log P(Z_j \mid \mathcal{H}_j)$ gives

$$\begin{aligned} I(G; Z_j \mid \mathcal{H}_j) = \mathbb{E} D_{\mathrm{KL}}(P(Z_j \mid G, \mathcal{H}_j) \| Q_j) \\ - \mathbb{E} D_{\mathrm{KL}}(P(Z_j \mid \mathcal{H}_j) \| Q_j). \end{aligned} \tag{9}$$

Summing over all layer/time updates bounds $I(G; G_e)$ when G_e is a readout solely of those random states. This compression bound concerns the discrete stochastic encoder; Concrete training and mean inference are computational surrogates. The priors are Bernoulli and Gaussian, with a nondegenerate feature posterior. For a normalized predictor q_ψ, conditional KL similarly gives $I(Y; G_e) \geq H(Y) + \mathbb{E} \log q_\psi(Y \mid G_e)$ for discrete labels $Y = G'$ (Alemi et al., 2017; Barber & Agakov, 2003). Since $H(Y)$ is parameter-independent, we use $\widehat{I}_{\mathrm{pred}} = \mathbb{E} \log q_\psi(Y \mid G_e)$. Thus, with $\lambda > 0$,

$$\mathcal{L} = \mathcal{U}_{\mathrm{GIB}} - \lambda \left(\mathrm{SMI}(G; G_e) + \mathrm{FMI}(G; G_e) \right). \tag{10}$$

Here λ weights the empirical objectives in equations 6 and 4. Formally, $\mathcal{L}_{\mathrm{GIB}} \leq \mathcal{U}_{\mathrm{GIB}} - H(Y)$; Appendix E.3 derives its nonnegative slack and the bounded regularized objective.

Proposition 3.1 (Task-information decomposition)**.** *Let Y be a finite-valued target, $D = (A, X)$ the observed input, and $Z = G_e$ a fixed stochastic encoder satisfying $Y \to D \to Z$. The input and representation may be continuous. The optimal conditional log-risk satisfies*

$$\begin{aligned} \mathcal{R}^*_{\log}(Z) = H(Y \mid Z) &= H(Y \mid D) + I(Y; D \mid Z) \\ &= H(Y \mid D) + I(Y; A \mid Z) + I(Y; X \mid A, Z). \end{aligned} \tag{11}$$

For any normalized predictor q, its excess over this optimal risk is $\mathbb{E}_Z D_{\mathrm{KL}}(P(Y \mid Z) \| q(Y \mid Z)) \geq 0$. Thus prediction error separates into intrinsic target uncertainty, discarded task information, and readout mismatch under the evaluation distribution.

Proof. Expand $I(Y; D \mid Z) = H(Y \mid Z) - H(Y \mid D, Z)$ and use the encoder Markov property to replace the latter entropy by $H(Y \mid D)$. The conditional mutual-information chain rule separates $D = (A, X)$. Adding and subtracting $\log P(Y \mid Z)$ in the expected log-loss gives the readout KL identity. These steps involve the entropy of the discrete target, so continuous attributes require no quantization.

For finite-support inputs, the metric connection is explicit: $H(A \mid Z) + H(X \mid Z) \leq H(A) + H(X) - I_{\mathrm{sp}} - J_F$, where J_F is the population FMI critic objective. Appendix E.3 proves this input-recovery bound and the resulting upper bound on task log-risk using local kernel preparation and NWJ data processing. Finite-precision records provide an implementable setting. The supervised term targets label information, while SMI/FMI reward recoverable input information; Appendix G.12 separately controls readout generalization.

4 Experiments

To evaluate the effectiveness of IDG, we seek to answer the following four questions. **Q1 (Downstream performance):** how does IDG compare with competitive baselines? **Q2 (Interpretability study):** how do SMI and FMI relate to latent embedding quality and downstream performance? **Q3 (Metric insights):** what insights follow from the correlation patterns? **Q4 (Model analysis):** what do ablations and robustness studies reveal about IDG across settings and scales?

4.1 Experimental setup

Datasets Link prediction and node classification use four standard datasets (GitHub, as well as Mooc, Wikipedia, and Reddit (Kumar et al., 2019)); link prediction additionally uses Tgbl-Wiki (Huang et al., 2023). Official node-affinity ranking uses Tgbn-Reddit, Tgbn-Genre, and Tgbn-Token with NDCG@10. For graph classification, we employ Dblp, Facebook, Highschool, Infectious, MIT, and Tumblr (Eagle & Pentland, 2006; Viswanath et al., 2009; Isella et al., 2011; Oettershagen et al., 2020). Timestamped events are processed chronologically into causal snapshot inputs. Appendix F.2 records dataset statistics and preprocessing; Appendix H.4 details official node-affinity ranking evaluation and additional graph benchmarks.

Table 1: Link prediction results (mean ± standard deviation, five runs). **Bold**/underline: best/second-best mean. $\Delta_b = 100(\bar{s}_{\mathrm{IDG}}/\bar{s}_b - 1)$ is the relative gain over baseline b in the five-dataset mean $\bar{s}$ of AUC or AP.

AUC performance						
Baseline	**Reddit**	**Mooc**	**GitHub**	**Wikipedia**	**Tgbl-Wiki**	$\Delta(\%)$
Jodie	0.732±0.007	0.731±0.008	0.750±0.005	0.755±0.005	0.738±0.004	↑15.4
DyRep	0.727±0.006	0.690±0.009	0.719±0.005	0.800±0.004	0.696±0.004	↑17.8
DySAT	0.730±0.007	0.673±0.009	0.608±0.007	0.728±0.006	0.882±0.003	↑18.1
DyGNN	0.741±0.008	0.675±0.010	0.641±0.008	0.623±0.007	0.861±0.004	↑20.8
TGN	0.746±0.005	0.741±0.006	0.719±0.005	0.848±0.003	0.782±0.004	↑11.5
PINT	0.732±0.006	0.806±0.005	0.644±0.007	0.753±0.005	0.908±0.003	↑11.3
DyGMamba	0.756±0.004	0.750±0.005	0.756±0.004	0.843±0.003	0.912±0.002	↑6.5
RAG4DyG	0.760±0.004	0.748±0.005	0.760±0.004	0.833±0.003	0.909±0.002	↑6.7
DyCIL	0.772±0.005	0.814±0.006	0.775±0.005	0.853±0.004	0.919±0.003	↑3.5
LAMP	0.779±0.004	0.799±0.006	0.769±0.005	0.861±0.004	0.926±0.003	↑3.5
IDG	**0.806±0.004**	**0.842±0.006**	**0.789±0.005**	**0.886±0.004**	**0.955±0.002**	-
AP performance						
Baseline	**Reddit**	**Mooc**	**GitHub**	**Wikipedia**	**Tgbl-Wiki**	$\Delta(\%)$
Jodie	0.634±0.008	0.682±0.008	0.687±0.006	0.764±0.005	0.857±0.004	↑18.4
DyRep	0.725±0.007	0.665±0.009	0.714±0.005	0.791±0.004	0.847±0.004	↑14.6
DySAT	0.728±0.006	0.659±0.009	0.576±0.008	0.771±0.005	0.866±0.003	↑19.2
DyGNN	0.749±0.007	0.625±0.011	0.641±0.008	0.623±0.007	0.891±0.003	↑21.6
TGN	0.767±0.005	0.735±0.006	0.734±0.005	0.870±0.003	0.837±0.004	↑8.8
PINT	0.744±0.006	0.763±0.005	0.630±0.008	0.758±0.005	0.905±0.003	↑12.9
DyGMamba	0.770±0.004	0.744±0.005	0.740±0.005	0.807±0.004	0.912±0.002	↑8.0
RAG4DyG	0.768±0.004	0.749±0.005	0.758±0.004	0.870±0.003	0.894±0.003	↑6.2
DyCIL	0.780±0.005	0.776±0.006	0.770±0.005	0.868±0.004	0.917±0.003	↑4.4
LAMP	0.789±0.005	0.767±0.006	0.766±0.005	0.883±0.004	0.930±0.003	↑3.7
IDG	**0.816±0.004**	**0.803±0.007**	**0.797±0.004**	**0.913±0.003**	**0.961±0.002**	-

Baselines We compare IDG against ten temporal models: TGN (Rossi et al., 2020), Jodie (Kumar et al., 2019), DyRep (Trivedi et al., 2019), DySAT (Sankar et al., 2020), the DyGNN baseline (Ma et al., 2020), PINT (Souza et al., 2022), DyGMamba (Ding et al., 2025), RAG4DyG (Wu et al., 2025), and two 2026 snapshot-based models, DyCIL (Zhang et al., 2026) and LAMP (De Castelli et al., 2026). For graph classification, kernel methods (Oettershagen et al., 2020) combine three representations (RD, DL, SE) with random walk (RW) or Weisfeiler-Lehman (WL) kernels; APPR-S is a complementary non-neural comparator for graph classification.

Assessment metrics For link prediction, we report AUC and average precision (AP). For node classification, we report AUC, AP, and recall. For graph classification, we report accuracy (ACC). Node-affinity ranking uses the official TGB NDCG@10 evaluator on the full destination set. To ensure fairness in node classification, all temporal models are assessed under the same protocol: every encoder is paired with the same lightweight classifier and optimized jointly with that head using training labels.

Implementation details Unless otherwise stated, IDG uses Adam, a 3-layer attention-based causal encoder (Veličković et al., 2018) with hidden width 128, dropout 0.1, and learning rate 10^{-4}. Training allows at most 1000 epochs with early-stopping patience of 100 epochs without validation improvement. Non-TGB link and node prediction use 5 independent runs and a 70:15:15 split; TGB retains official splits. All per-epoch times are measured on one NVIDIA RTX 3090 with 24 GB memory. Appendix F.1 specifies task settings and checkpoint selection.

For the event-stream benchmarks, we use fixed chronological windows before constructing snapshots; in particular, Tgbl-Wiki is processed with the windowing protocol described in Appendix G. The default weights are $\lambda = 0.01$, $\beta = 0.1$, and $k_{\mathrm{nbr}} = 10$, selected from the cross-task sensitivity study. The same lightweight readout is used for every temporal baseline in node classification. For Mooc, the unipartite event-role adaptation of Jodie is used so that the comparison does not depend on a missing bipartite user/item interface under this controlled protocol.

Table 2: Node classification (mean ± standard deviation, five runs). **Bold**/underline: best/second-best mean; Δ is the relative gain in the four-dataset mean, as defined in Table 1.

AUC performance					
Baseline	**Reddit**	**Mooc**	**GitHub**	**Wikipedia**	**Δ(%)**
Jodie	0.869±0.006	0.753±0.007	0.813±0.005	0.876±0.005	↑8.8
DyRep	0.751±0.008	0.711±0.008	0.767±0.006	0.796±0.007	↑19.1
DySAT	0.809±0.007	0.786±0.007	0.795±0.006	0.852±0.006	↑11.1
DyGNN	0.836±0.006	0.836±0.006	0.828±0.005	0.780±0.008	↑9.8
TGN	0.717±0.008	0.752±0.008	0.787±0.006	0.867±0.005	↑15.4
PINT	0.827±0.006	0.809±0.006	0.808±0.005	0.878±0.005	↑8.5
DyGMamba	0.830±0.005	0.828±0.005	0.822±0.005	0.880±0.004	↑7.2
RAG4DyG	0.844±0.005	0.840±0.005	0.830±0.005	0.878±0.005	↑6.2
DyCIL	0.878±0.005	0.853±0.006	0.847±0.005	0.897±0.005	↑3.7
LAMP	0.876±0.005	0.848±0.005	0.844±0.005	0.904±0.004	↑3.8
IDG	**0.908±0.005**	**0.883±0.005**	**0.876±0.005**	**0.936±0.004**	-
AP performance					
Baseline	**Reddit**	**Mooc**	**GitHub**	**Wikipedia**	**Δ(%)**
Jodie	0.641±0.009	0.680±0.008	0.722±0.006	0.755±0.006	↑15.7
DyRep	0.612±0.010	0.667±0.009	0.617±0.009	0.737±0.007	↑23.0
DySAT	0.677±0.008	0.683±0.008	0.731±0.007	0.778±0.006	↑12.9
DyGNN	0.685±0.008	0.685±0.008	0.734±0.006	0.812±0.006	↑11.0
TGN	0.631±0.010	0.615±0.010	0.755±0.006	0.829±0.005	↑14.4
PINT	0.718±0.007	0.674±0.008	0.778±0.005	0.796±0.006	↑9.2
DyGMamba	0.739±0.006	0.703±0.007	0.798±0.005	0.801±0.005	↑6.5
RAG4DyG	0.727±0.006	0.694±0.007	0.784±0.005	0.808±0.005	↑7.5
DyCIL	0.748±0.006	0.722±0.007	0.809±0.006	0.837±0.005	↑3.9
LAMP	0.756±0.006	0.719±0.007	0.805±0.005	0.843±0.005	↑3.7
IDG	**0.782±0.005**	**0.747±0.005**	**0.837±0.005**	**0.872±0.004**	-
Recall performance					
Baseline	**Reddit**	**Mooc**	**GitHub**	**Wikipedia**	**Δ(%)**
Jodie	0.686±0.008	0.782±0.007	0.689±0.007	0.637±0.008	↑16.1
DyRep	0.767±0.007	0.761±0.007	0.761±0.006	0.660±0.008	↑10.0
DySAT	0.720±0.008	0.675±0.009	0.705±0.007	0.629±0.009	↑18.8
DyGNN	0.689±0.009	0.789±0.007	0.660±0.008	0.614±0.009	↑17.8
TGN	0.752±0.008	0.793±0.007	0.722±0.007	0.592±0.010	↑13.4
PINT	0.739±0.007	0.764±0.007	0.739±0.006	0.692±0.007	↑10.5
DyGMamba	0.752±0.006	0.778±0.006	0.768±0.006	0.729±0.006	↑7.1
RAG4DyG	0.760±0.006	0.771±0.006	0.760±0.006	0.692±0.007	↑8.7
DyCIL	0.778±0.007	0.809±0.006	0.784±0.006	0.745±0.007	↑4.1
LAMP	0.785±0.006	0.804±0.006	0.779±0.006	0.757±0.006	↑3.8
IDG	**0.812±0.006**	**0.837±0.005**	**0.811±0.006**	**0.783±0.005**	-

4.2 Downstream performance (Q1)

Obs 1. IDG achieves the highest mean AUC and AP on all five link-prediction datasets. Against the strongest baseline in each column of Table 1, IDG gains 0.014 to 0.029 AUC and 0.027 to 0.031 AP. On Mooc, its AUC/AP is 0.842/0.803, exceeding DyCIL's 0.814/0.776. The five AUC gains are 0.027, 0.028, 0.014, 0.025, and 0.029, respectively, across the five datasets under the shared chronological prediction and evaluation protocol.

Obs 2. IDG obtains the highest mean AUC, AP, and recall on all four node-classification datasets. In Table 2, its gains over the strongest baseline range from 0.029 to 0.032 AUC, 0.025 to 0.029 AP, and 0.026 to 0.028 recall. DyCIL and LAMP provide the strongest competing means throughout this table. These results complement the objective ablations in Appendix H, which examine each objective term under the same experimental protocol.

Official node-affinity ranking. IDG ranks first on Tgbn-Genre and Tgbn-Token, with NDCG@10 of 0.406 and 0.269 (Table 18). These exceed the strongest baseline, LAMP, by 0.014 and 0.011. On Tgbn-Reddit, IDG ranks third at 0.338, behind LAMP (0.348) and DyCIL (0.340). Thus, the ranking gains are dataset-dependent and complement the binary-classification results.

4.3 INTERPRETABILITY STUDY (Q2)

We analyze the Pearson correlations of SMI and FMI with link-prediction AUC across IDG and all baselines on five datasets. Appendix G presents the complete visualization and the corresponding scores for each of the five evaluated dynamic graph datasets.

Obs 3. Both metrics show positive empirical association with model performance. Across the eleven-model comparison, SMI and FMI achieve average R^2 values of 0.840 and 0.855, respectively. These are unweighted averages of five dataset-wise squared Pearson correlations, computed from the displayed AUC and normalized metric values. Both metrics outperform all five intrinsic controls in both correlations on every dataset (Appendix G.6); Appendix G.8 examines distribution shift.

Obs 4. Direct optimization of SMI and FMI improves the evaluated latent embeddings. The ablations in Appendix H support combining both metrics with task supervision, while Appendix G examines their behavior under distribution shift.

4.4 METRIC INSIGHTS (Q3)

Obs 5. The two metrics provide complementary assessment coordinates. SMI quantifies structural information shared by inputs and latent embeddings, while FMI measures attribute and latent embedding dependence. Together, they distinguish structural and feature changes under controlled edge rewiring and attribute permutation of the observed graph inputs. At 20% perturbation, rewiring lowers SMI by 35.60% to 49.70%, and attribute permutation lowers FMI by 17.10% to 28.80%, with smaller off-target changes across five datasets (Appendix G.11).

Obs 6. Metric associations depend on the dataset. SMI has the higher R^2 on Mooc and GitHub, whereas FMI has the higher R^2 on Reddit, Wikipedia, and Tgbl-Wiki; the largest gap is 0.121. These patterns motivate selecting metric weights on validation data for each application. The controlled sensitivity studies test regularization strength separately from this cross-model analysis.

4.5 MODEL ANALYSIS (Q4)

Obs 7. Ablations support all three objective components. Removing SMI, FMI, or the bottleneck lowers performance, while structural and feature perturbations selectively reduce the corresponding metric. Appendix H provides the complete plots and extended tests.

Obs 8. Controlled studies examine estimator, distribution-shift, and temporal-resolution effects. With known Gaussian mutual information, more samples stabilize FMI, while critic capacity separates approximation from sample-and-optimization error (Appendix G.7). DIDA shifts reduce the corresponding structural or feature metric and weaken pooled correlations (Appendix G.8). On Tgbl-Wiki, windows of 1 to 300 seconds are stable, whereas 900-second windows lower performance under the joint change in aggregation and physical history covered (Appendix G.9).

Excluding IDG, the ten baselines retain average SMI/FMI $R^2 = 0.848/0.832$ (Appendix G.10), supporting the same positive association across independently trained architectures.

IDG suits window-based prediction with changing neighborhoods and attributes. Official node-affinity ranking tests extend it to large interaction streams: IDG attains the highest NDCG@10 on Tgbn-Genre and Tgbn-Token, and combines third-highest NDCG@10 with second-lowest training time per epoch on Tgbn-Reddit under the shared timing protocol.

5 CONCLUSIONS

We propose **IDG**, combining SMI, FMI, and a task-aware bottleneck for dynamic graph assessment and learning. Experiments support complementary structural and feature diagnostics. A task-information decomposition and finite-sample bound characterize discarded task information and readout error; Appendices A.1 to G provide protocols and proofs. The shared assessment interface applies to IDG and baseline latent embeddings. Correlation controls and targeted perturbations connect both scores to downstream utility and distinguish structural from attribute degradation, supporting interpretable evaluation of dynamic graph representations.

AI USE STATEMENT

Generative AI assisted with language editing, consistency checks, and typesetting. The authors provided the research background, experimental methods, experimental measurements, figures, tables, and data analysis. Derived gains and correlations were calculated from those measurements. The authors retain full responsibility for all content.

ETHICS STATEMENT

This study evaluates dynamic graph learning on public benchmarks. The proposed diagnostics support transparent assessment of structural and feature information in latent embeddings and can help identify changes in model behavior. Social interaction data warrant particular attention to privacy, access conditions, and potential misuse. Responsible use should respect dataset licenses, protect individual identities, and avoid attempts to infer sensitive personal information. Applications involving people should include task-specific validation, assessment of disparate outcomes, and appropriate human oversight. These practices complement the technical evaluation when adapting IDG to real-world settings.

REPRODUCIBILITY STATEMENT

To promote transparency and reproducibility, we include the source code of IDG, together with training and evaluation scripts, in the anonymous repository at `https://anonymous.4open.science/r/IDG_ICLR-924F`. We also mirror the essential implementation files in the supplementary material for review during anonymous evaluation. The probability model and estimators are specified in Appendix A.7, and the experimental protocols are recorded in Appendix F.1. We welcome researchers to reproduce, audit, adapt, and extend this work across dynamic graph benchmarks.

REFERENCES

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

David Barber and Felix V. Agakov. Information maximization in noisy channels: A variational approach. In *Advances in Neural Information Processing Systems*, volume 16, 2003. URL `https://papers.nips.cc/paper_files/paper/2003/hash/a6ea8471c120fe8cc35a2954c9b9c595-Abstract.html`.

Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere. The million song dataset, 2011. Dataset release.

Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 3438–3445, 2020.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2 edition, 2006.

Fabrizio De Castelli, Alessio Gravina, Moshe Eliasof, Carola-Bibiane Schönlieb, and Davide Bacciu. Adaptive memory retention in dynamic graphs. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL `https://openreview.net/forum?id=x7NGYPNlgs`.

Zifeng Ding, Yifeng Li, Yuan He, Antonio Norelli, Jingcheng Wu, Volker Tresp, Michael Bronstein, and Yunpu Ma. Dygmamba: Efficiently modeling long-term temporal dependency on continuous-time dynamic graphs with state space models, 2025. URL `https://arxiv.org/abs/2408.04713`.

Nathan Eagle and Alex Pentland. Reality mining: sensing complex social systems. *Personal and ubiquitous computing*, 10(4):255–268, 2006.

ZhengZhao Feng, Rui Wang, TianXing Wang, Mingli Song, Sai Wu, and Shuibing He. A comprehensive survey of dynamic graph neural networks: Models, frameworks, benchmarks, experiments and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 38(1):26–46, 2026. doi: 10.1109/TKDE.2025.3621291.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann Lecun. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International conference on machine learning*, pp. 10929–10974. PMLR, 2023.

Julia Gastinger, Shenyang Huang, Mikhail Galkin, Erfan Loghmani, Ali Parviz, Farimah Poursafaei, Jacob Danovitch, Emanuele Rossi, Ioannis Koutis, Heiner Stuckenschmidt, Reihaneh Rabbany, and Guillaume Rabusseau. TGB 2.0: A benchmark for learning on temporal knowledge graphs and heterogeneous graphs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 140199–140229. Curran Associates, Inc., 2024. doi: 10.52202/079017-4450.

Arthur Gretton, Olivier Bousquet, Alexander Smola, and Bernhard Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *Algorithmic Learning Theory*, volume 3734 of *Lecture Notes in Computer Science*, pp. 63–77. Springer, 2005. doi: 10.1007/11564089_7. URL https://www.gatsby.ucl.ac.uk/~gretton/papers/GreBouSmoSch05.pdf.

Lin Han, Richard C Wilson, and Edwin R Hancock. Generative graph prototypes from information theory. *IEEE transactions on pattern analysis and machine intelligence*, 37(10):2013–2027, 2015.

Fengrui Hua, Yiyan Qi, Zikai Wei, Yuxing Tian, Chengjin Xu, Xiaojun Wu, Jia Li, and Jian Guo. BAG: Benchmarking anomaly detection on dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 14892–14900, 2026. doi: 10.1609/aaai.v40i17.38510.

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*, volume 36, 2023. doi: 10.52202/075280-0099.

Lorenzo Isella, Juliette Stehlé, Alain Barrat, Ciro Cattuto, Jean-François Pinton, and Wouter Van den Broeck. What's in a crowd? analysis of face-to-face behavioral networks. *Journal of theoretical biology*, 271(1):166–180, 2011.

Pengfei Jiao, Hongjiang Chen, Xuan Guo, Zhidong Zhao, Dongxiao He, and Di Jin. A survey on temporal interaction graph representation learning: Progress, challenges, and opportunities. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 10499–10507. International Joint Conferences on Artificial Intelligence Organization, 2025. doi: 10.24963/ijcai.2025/1166. URL https://doi.org/10.24963/ijcai.2025/1166.

Yannis Karmim, Marc Lafon, Raphaël Fournier-S'niehotta, and Nicolas Thome. Supra-laplacian encoding for transformer on dynamic graphs. *Advances in Neural Information Processing Systems*, 37:17215–17246, 2024.

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 & data mining*, pp. 1269–1278, 2019.

Jintang Li, Ruofan Wu, Xinzhou Jin, Boqun Ma, Liang Chen, and Zibin Zheng. State space models on temporal graphs: A first-principles study. In *Advances in Neural Information Processing Systems*, volume 37, pp. 127030–127058. Curran Associates, Inc., 2024. doi: 10.52202/079017-4034.

Katharina Limbeck, Rayna Andreeva, Rik Sarkar, and Bastian Rieck. Metric space magnitude for evaluating the diversity of latent representations. *Advances in Neural Information Processing Systems*, 37:123911–123953, 2024.

Jingyu Liu, Huayi Tang, and Yong Liu. Perfect alignment may be poisonous to graph contrastive learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 31890–31916. PMLR, 2024. URL https://proceedings.mlr.press/v235/liu24bd.html.

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

Yao Ma, Ziyi Guo, Zhaocun Ren, Jiliang Tang, and Dawei Yin. Streaming graph neural networks. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pp. 719–728, 2020.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

Alessio Micheli and Domenico Tortorella. Discrete-time dynamic graph echo state networks. *Neurocomputing*, 496:85–95, 2022. doi: 10.1016/j.neucom.2022.05.001.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, 2 edition, 2018.

Amirhossein Nadiri and Frank W Takes. A large-scale temporal analysis of user lifespan durability on the reddit social media platform. In *Companion Proceedings of the Web Conference 2022*, pp. 677–685, 2022.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

Lutz Oettershagen, Nils M Kriege, Christopher Morris, and Petra Mutzel. Temporal graph kernels for classifying dissemination processes. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pp. 496–504. SIAM, 2020.

Ashwin Paranjape, Austin R Benson, and Jure Leskovec. Motifs in temporal networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, pp. 601–610, 2017.

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5363–5370, 2020.

Ben Poole, Sherjil Ozair, Aaron van den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5171–5180, 2019. URL `https://proceedings.mlr.press/v97/poole19a.html`.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*, 2020.

Luis Gonzalo Sanchez Giraldo, Murali Rao, and Jose C. Principe. Measures of entropy from data using infinitely divisible kernels. *IEEE Transactions on Information Theory*, 61(1):535–548, 2015. doi: 10.1109/TIT.2014.2370058.

Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th international conference on web search and data mining*, pp. 519–527, 2020.

Naw Safrin Sattar, Khaled Z Ibrahim, Aydin Buluc, and Shaikh Arifuzzaman. Dyg-dpcd: A distributed parallel community detection algorithm for large-scale dynamic graphs. *International Journal of Parallel Programming*, 53(1):4, 2025.

Kiarash Shamsi, Friedhelm Victor, Murat Kantarcioglu, Yulia Gel, and Cuneyt G Akcora. Chartalist: Labeled graph datasets for utxo and account-based blockchains. *Advances in Neural Information Processing Systems*, 35:34926–34939, 2022.

Amauri Souza, Diego Mesquita, Samuel Kaski, and Vikas Garg. Provably expressive temporal graph networks. *Advances in neural information processing systems*, 35:32257–32269, 2022.

Junwei Su and Shan Wu. Temporal-aware evaluation and learning for temporal graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20628–20636, 2025. doi: 10.1609/aaai.v39i19.34273.

Katherine Tieu, Dongqi Fu, Yada Zhu, Hendrik Hamann, and Jingrui He. Temporal graph neural tangent kernel with graphon-guaranteed. In *Advances in Neural Information Processing Systems*, volume 37, pp. 94568–94606. Curran Associates, Inc., 2024. doi: 10.52202/079017-2998.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *International conference on learning representations*, 2019.

Md Joshem Uddin, Soham Changani, and Baris Coskunuzer. T3former: Temporal graph classification with topological machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26028–26036, 2026. doi: 10.1609/aaai.v40i31.39804.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations*, 2019.

Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P Gummadi. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM workshop on Online social networks*, pp. 37–42, 2009.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pp. 9929–9939. PMLR, 2020.

John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018.

Shirley Wu, Kaidi Cao, Bruno Ribeiro, James Zou, and Jure Leskovec. GraphMETRO: Mitigating complex graph distribution shifts via mixture of aligned experts. In *Advances in Neural Information Processing Systems*, volume 37, pp. 9358–9387. Curran Associates, Inc., 2024. doi: 10.52202/079017-0297.

Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. Graph information bottleneck. *Advances in Neural Information Processing Systems*, 33:20437–20448, 2020.

Yuxia Wu, Lizi Liao, and Yuan Fang. Retrieval augmented generation for dynamic graph modeling. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1434–1443. ACM, 2025. doi: 10.1145/3726302.3729958. URL https://doi.org/10.1145/3726302.3729958.

Kuo Yang, Zhengyang Zhou, Qihe Huang, Limin Li, Yuxuan Liang, and Yang Wang. Improving generalization of dynamic graph learning via environment prompt. In *Advances in Neural Information Processing Systems*, volume 37, pp. 70048–70075. Curran Associates, Inc., 2024. doi: 10.52202/079017-2239.

Nan Yin, Mengzhu Wang, Zhenghan Chen, Giulia De Masi, Huan Xiong, and Bin Gu. Dynamic spiking graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16495–16503, 2024. doi: 10.1609/aaai.v38i15.29587.

Shujian Yu, Francesco Alesiani, Xi Yu, Robert Jenssen, and Jose Principe. Measuring dependence with matrix-based entropy functional. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10781–10789, 2021. doi: 10.1609/aaai.v35i12.17288.

Haonan Yuan, Qingyun Sun, Xingcheng Fu, Cheng Ji, and Jianxin Li. Dynamic graph information bottleneck. In *Proceedings of the ACM Web Conference 2024*, pp. 469–480. ACM, 2024. doi: 10.1145/3589334.3645411. URL https://doi.org/10.1145/3589334.3645411.

Haonan Yuan, Qingyun Sun, Zhaonan Wang, Xingcheng Fu, Cheng Ji, Yongjian Wang, Bo Jin, and Jianxin Li. DG-Mamba: Robust and efficient dynamic graph structure learning with selective state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22272–22280, 2025. doi: 10.1609/aaai.v39i21.34382.

Jiasheng Zhang, Jialin Chen, Menglin Yang, Aosong Feng, Shuang Liang, Jie Shao, and Rex Ying. DTGB: A comprehensive benchmark for dynamic text-attributed graphs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 91405–91429. Curran Associates, Inc., 2024. doi: 10.52202/079017-2901.

Xinxun Zhang, Pengfei Jiao, Mengzhou Gao, Tianpeng Li, and Xuan Guo. Towards OOD generalization in dynamic graphs via causal invariant learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 16379–16387, 2026. doi: 10.1609/aaai.v40i19.38676.

Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Zhou Qin, and Wenwu Zhu. Dynamic graph neural networks under spatio-temporal distribution shift. In *Advances in Neural Information Processing Systems*, volume 35, pp. 6074–6089, 2022.

A FREQUENTLY ASKED QUESTIONS (FAQS)

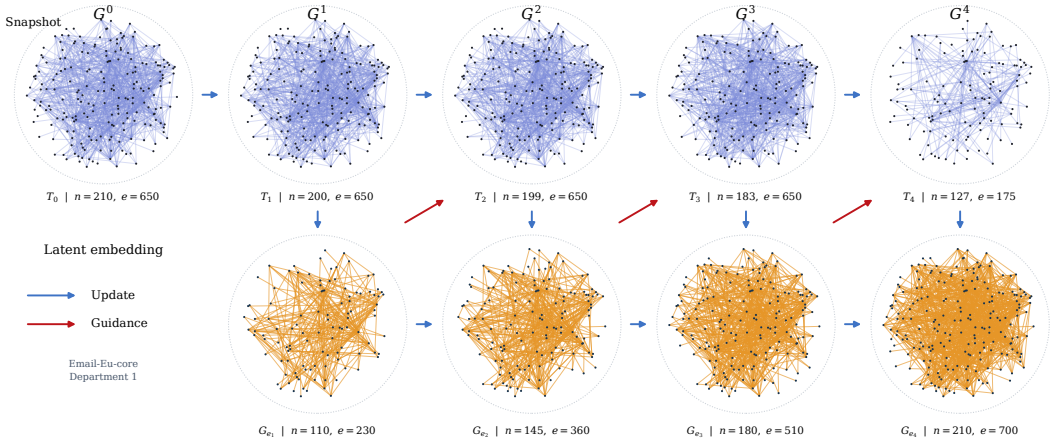

Figure 3: Snapshot-driven latent embedding evolution illustrated on an Email-Eu-core Department 1 sample. Each input snapshot follows the half-open window convention in Appendix G.1. Successive snapshots update $G_{e_1}, \ldots, G_{e_4}$ with progressively larger node and edge sets.

This section collects the most common reproducibility, metric-design, and complexity questions so that the supplementary derivations and experiments can be read with the intended assumptions in place before the detailed answers are used in the following sections.

A.1 IMPLEMENTATION DETAILS

Algorithm 1 and Appendix A.7 specify the causal encoder, structural gates, Gaussian feature states, prediction head, and metric estimators. Appendix F.1 records the training configuration, while Appendix F.2 defines data preprocessing and chronological evaluation. Together these components specify how the graph sequence enters each update and how the resulting states are assessed.

A.2 WHY DO WE CHOOSE SPECTRAL ENTROPY FOR SMI?

We use spectral entropy because it yields a joint input and latent embedding information measure from normalized positive semidefinite operators. Under paired-record and normalized positive-definite-kernel construction, the joint and marginal operators have consistent spectra and a nonnegative mutual information (Sanchez Giraldo et al., 2015; Watrous, 2018). Their Gram matrices give a finite-dimensional computation from sparse structural records and latent embeddings. Appendix D.1 proves the entropy formula and its bounds.

A.3 WHY DO WE ADOPT f-DIVERGENCE-BASED REPRESENTATION FOR FMI?

We use the f-divergence representation to estimate FMI with a finite neural function class. The variational form provides a computable lower bound, with approximation, sampling, and optimization errors characterized separately. Joint pairs preserve attribute and latent embedding alignment, and independently sampled marginal pairs provide the reference distribution for estimating the dependence of attributes and latent embeddings.

A.4 TIME COMPLEXITY ANALYSIS OF IDG

We analyze IDG using sparse neighborhoods and bounded-size paired spectral assessment.

Algorithm 1 provides the training procedure for the operation count. Table 3 reports the corresponding per-epoch runtimes across five link-prediction datasets on one NVIDIA RTX 3090.

Update order. Algorithm 1 orders updates by time and then layer. Each structural draw precedes its feature draw, so the conditioning history contains every state used by the next update.

Table 3: Average training time per epoch (in seconds) for link prediction on five datasets. Values are reported as mean (± standard deviation). **Bold**/underline: best/second-best mean (lower is better). Timing was measured on one NVIDIA RTX 3090.

Method	Reddit	Mooc	GitHub	Wikipedia	Tgbl-Wiki
Jodie	74.35 (±2.65)	52.73 (±0.22)	**4.82 (±0.02)**	43.30 (±0.48)	2,645.33 (±58.21)
DyRep	111.83 (±3.69)	64.58 (±1.02)	6.72 (±0.18)	30.30 (±0.81)	2,796.94 (±65.93)
DySAT	11.34 (±1.47)	43.41 (±1.42)	31.21 (±1.84)	**3.77 (±0.54)**	425.18 (±10.55)
DyGNN	1,891.73 (±30.04)	646.62 (±12.00)	85.49 (±2.66)	487.58 (±7.29)	48,520.50 (±1,250.12)
TGN	81.25 (±2.11)	54.12 (±2.51)	5.55 (±0.08)	17.15 (±0.16)	3,171.64 (±687.25)
PINT	1,016.64 (±11.00)	527.88 (±16.81)	7.37 (±0.14)	62.48 (±0.77)	28,450.60 (±680.33)
DyGMamba	152.40 (±5.20)	82.65 (±3.10)	6.80 (±0.12)	25.30 (±1.20)	5,120.80 (±150.30)
RAG4DyG	123.70 (±4.80)	65.40 (±2.80)	6.20 (±0.10)	20.50 (±1.00)	4,050.20 (±120.50)
DyCIL	28.60 (±1.24)	96.40 (±3.85)	58.70 (±2.10)	12.80 (±0.56)	1,148.60 (±42.70)
LAMP	17.80 (±0.76)	65.20 (±2.48)	42.60 (±1.58)	7.90 (±0.34)	682.40 (±24.30)
IDG (Ours)	**7.65 (±0.45)**	**40.17 (±0.75)**	37.75 (±1.04)	6.77 (±0.36)	**385.42 (±6.13)**

Algorithm 1 IDG

Input: Training prefixes $G^{1:T}$ and labels Y; layers L; neighbors k_{nbr}; horizon Δt; weights λ, β; paired-record budget $b = 128$; relaxation temperature $\tau_r = 0.5$.
Output: Encoder state G_e and normalized predictor $q_\psi(Y \mid G_e)$.

Initialize encoder ω, critic θ, and decoder ψ; fix priors Q_A, Q_X, persistent node indices, record budget b, and kernel bandwidths.
for each training iteration **do**
  Select the next block of at most $W = 32$ observed windows and its targets; carry detached historical states (zero at sequence start); set $H^{0,t} = X_t$.
  for each window t in the selected block **do**
    for $l = 1, \ldots, L$ **do**
      Form causal candidate neighbors and $C_v^{l,t}$; compute gate probabilities $\phi_{vu}^{l,t}$ (equation 15).
      Draw binary Concrete gates $\widetilde{M}^{l,t}$ (equation 14); accumulate discrete-prior KL $\mathcal{K}_A^{l,t}$ (equation 16).
      Aggregate $R_v^{l,t}$; compute Gaussian mean $m_v^{l,t}$ and positive scale $s_v^{l,t}$ (equation 17).
      Draw $\xi_v^{l,t} \sim \mathcal{N}(0, I)$ and set $H_v^{l,t} = m_v^{l,t} + s_v^{l,t} \odot \xi_v^{l,t}$; accumulate $\mathcal{K}_X^{l,t}$ (equation 18).
      Retain paired observed structural descriptors $a_{v,t}$ and latent means $z_{v,t}^l = m_v^{l,t}$ (equation 29).
    end for
  end for
  Draw b paired node/window records; form B_A, B_Z^l, B_{AZ}^l and sum their structure mutual information across layers (equations 3 to 4).
  Read out G_e from the retained states and compute $q_\psi(Y \mid G_e)$; form joint and independent feature pairs as in Section 3.2.2.
  With encoder outputs detached, take one Adam update of θ by ascent on the training NWJ objective $\widehat{J}_\theta$ (equation 6).
  With θ fixed, recompute the critic on differentiable encoder outputs; update ω, ψ by descent on $-\log q_\psi(Y \mid G_e) + \beta \sum_{l,t}(\mathcal{K}_A^{l,t} + \mathcal{K}_X^{l,t}) - \lambda(\mathrm{SMI} + \widehat{J}_\theta)$ (equation 10).
end for
Freeze parameters; predict with posterior means; assess final-layer SMI on shared paired records and FMI on disjoint critic-assessment records.

Critic ascent and encoder descent operate on separate parameter blocks; equation 9 applies to the complete history of these causally ordered latent updates.

Sparse encoder operations. Let n be the number of stored node states, T the number of processed windows, L the layer count, and d_h the hidden width. Let c_t count candidate edges at window t, so $c_t \le n k_{\mathrm{nbr}}$ for directed candidates. Feature transformations cost $O(LTnd_h^2)$ in a full-node implementation; candidate scoring and aggregation cost $O(L\sum_t c_t d_h)$. Gaussian and Bernoulli penalties add $O(LTnd_h)$ and $O(L\sum_t c_t)$, respectively. Excluded-edge contributions in equation 16 are obtained from counts. Active-node updates can further reduce this conservative accounting.

Paired spectral assessment. SMI uses a budget of $b \leq 128$ paired records per objective block. Let e_j count nonzeros in the sampled structural rows of block j and $J_{\mathrm{blk}} = \lceil T/W \rceil$ for a maximum of W windows per block. Computing all sparse-row inner products costs at most $O(be_j)$: each coordinate contributes at most b times its incidence count in the selected rows. Latent pairwise distances cost $O(Lb^2 d_h)$, and spectral entropy acts on three $b \times b$ matrices per layer with cost $O(Lb^3)$. The joint kernel is formed entrywise, so it has the same matrix dimension as each marginal. One objective pass therefore costs

$$C_{\mathrm{pass}} = O\Big(L \sum_{t=1}^{T} [n d_h^2 + c_t d_h] + \sum_{j=1}^{J_{\mathrm{blk}}} be_j + LJ_{\mathrm{blk}}(b^2 d_h + b^3) + C_{\mathrm{FMI}}(n_F) + C_{\mathrm{pred}}\Big). \tag{12}$$

Here n_F is the total number of sampled feature pairs; C_{FMI} includes critic fitting and C_{pred} includes the scored prediction examples. This bound includes backward differentiation up to constant factors and makes the batch-level entropy cost explicit.

Sparse-graph scaling and setup. Let $m_t = \mathrm{nnz}(A_t)$ count observed input nonzeros, which can exceed the candidate-edge count. Symmetrization contributes at most $2m_t$ entries. Even with repeated record draws, the fixed budget gives $\sum_j e_j = O(b \sum_t m_t + bJ_{\mathrm{blk}})$, including degree coordinates. With fixed $L, d_h, k_{\mathrm{nbr}}, b, W$ and bounded per-node prediction and critic sampling, the pass cost simplifies to $O(Tn + \sum_t m_t)$. For sparse snapshots with $m_t = O(n)$, this is $O(Tn)$. The entropy matrix dimension is determined by the record budget, while graph-size dependence enters through sparse input access and encoder updates.

Preprocessing cost C_{index} includes chronological ordering, snapshot assembly, and cached candidate lists. For N_{pass} optimization passes, total cost is $C_{\mathrm{index}} + N_{\mathrm{pass}} C_{\mathrm{pass}}$. The bound retains the number of processed windows and passes explicitly. Per-epoch measurements in Table 3 characterize the reported workloads, including window assembly, local updates, and metric estimation.

Memory and temporal blocking. We carry node states across chronological blocks and truncate gradients at block boundaries. With $W = 32$ windows, stored spectral matrices require $O(Lb^2)$ memory, independent of the full sequence length. Encoder activations require at most $O(LWnd_h + L \sum_{t \in \mathrm{block}} c_t d_h)$ in addition to persistent node states, sparse descriptors, and optimizer state. State carry preserves information from earlier windows while the bounded gradient horizon controls memory. This is a streaming recurrence with blockwise training, and the full-trajectory information identity conditions on the carried history.

We report average per-epoch training time on one NVIDIA RTX 3090. Relative to the fastest non-IDG baseline on each dataset, IDG takes $0.67\times$, $0.93\times$, $7.83\times$, $1.80\times$, and $0.91\times$ the time on Reddit, Mooc, GitHub, Wikipedia, and Tgbl-Wiki, respectively. It is faster than both DyCIL and LAMP on all five datasets, and fastest overall on Reddit, Mooc, and Tgbl-Wiki. It is second-fastest on Wikipedia but slower than several models on GitHub. These measured trade-offs depend on the dataset and computational workload.

A.5 Latent embedding update

The temporary latent embedding $\hat{G}_{e_t}^{l}$ is generated by the stochastic update operator implemented in Algorithm 1. Abstractly, we write the induced stochastic update compactly as

$$\hat{G}_{e_t}^{l} = \mathcal{U}_\omega(G_e^{l,t-1}, G_e^{l-1,t}; G_t), \tag{13}$$

where $\mathcal{U}_\omega$ is the encoder update with parameters ω. It contains four steps: a linear projection of snapshot features, retention probabilities over the local spatio-temporal neighborhood, Bernoulli sampling of latent structural connections, and aggregation followed by the stochastic feature head in equation 17. This notation makes explicit how each snapshot contributes to the intermediate latent embedding that later enters the SMI objective.

A.6 Relaxed Bernoulli sampling

For differentiable training, let a be a gate logit and define $\phi = \mathrm{clip}(\sigma(a), 0.01, 0.99)$. We use the reparameterized binary Concrete relaxation during training (Maddison et al., 2017):

$$\widetilde{b} = \sigma\left(\frac{\mathrm{logit}(\phi) + \log u - \log(1-u)}{\tau_r}\right), \qquad u \sim \mathrm{Uniform}(0,1), \quad \tau_r > 0. \tag{14}$$

We fix $\tau_r = 0.5$ throughout training. We use the reparameterized sampler so gradients pass through $\widetilde{b}$, ϕ, and a. Prediction uses the mean gate ϕ and Gaussian mean m. The finite-temperature relaxation gives a biased pathwise surrogate for the discrete objective; the discrete Bernoulli KL regularizes the gate probabilities. FMI regularization uses sampled feature states; cross-model assessment uses each model's latent embedding at inference, as specified in Appendix A.7.

A.7 Encoder specification and probability model

We combine a Bernoulli structural bottleneck with a nondegenerate Gaussian feature bottleneck, following variational information-bottleneck formulations (Wu et al., 2020; Alemi et al., 2017). The causal state, posterior distributions, sampling rules, and metric estimators below define the encoder in Algorithm 1 and the probability model used by the information bounds.

Causal state and local context. For prediction after time t, only $G^{1:t} = (G_1, \ldots, G_t)$ is available. A fixed padding basis indexes observed vertices, and every populated entry uses information available at the prediction cutoff. Let $H^{0,t} = X_t$, and initialize $H^{l,0}$ to zero. For layer l and node v, define the projected context

$$C_v^{l,t} = \mathrm{ELU}\big([H_v^{l-1,t} \| H_v^{l,t-1}] W_C^l + b_C^l\big), \quad \phi_{vu}^{l,t} = \mathrm{clip}_{[\epsilon_p, 1-\epsilon_p]}\, \sigma\big((a_{\mathrm{gate}}^l)^\top [C_v^{l,t} \| C_u^{l,t}] + b_{\mathrm{gate}}^l\big). \tag{15}$$

Here $\mathrm{ELU}(x) = x$ for $x \geq 0$ and $e^x - 1$ otherwise, $\sigma(x) = (1+e^{-x})^{-1}$, $\|$ concatenates vectors, and $W_C^l, b_C^l, a_{\mathrm{gate}}^l, b_{\mathrm{gate}}^l$ are trainable parameters. The candidate set $\mathcal{N}_{k_{\mathrm{nbr}},\Delta t}(v)$ contains at most k_{nbr} neighbors observed at windows t' with $t - \Delta t < t' \leq t$. We use a look-back horizon of $\Delta t = 32$ windows, retaining the most recent eligible neighbors and breaking equal-time ties by node index. We set $\epsilon_p = 0.01$; each projection maps its concatenated input to hidden width d_h. Here the cache horizon counts processed snapshots; timestamp membership follows the half-open intervals in Appendix G.1.

We align snapshots by persistent node identities: each observed row of S^t selects its common latent slot, with consistent padding for missing vertices. Equation 2 defines the structural reconstruction criterion motivating this graph representation. Training minimizes equation 10. All information bounds condition on the fixed schema, horizon, and encoder parameters, which are independent of future evaluation records.

Structural posterior and relaxation. Condition on a fixed vertex schema and a fixed potential edge universe $\mathcal{U}$ before defining the information quantities. The discrete posterior is $P_A^{l,t} = \prod_{(v,u)\in\mathcal{U}} \mathrm{Bern}(\phi_{vu}^{l,t})$, with $\phi_{vu}^{l,t} = 0$ outside the causal candidate set $\mathcal{C}_t$. The full-support prior is $Q_A^{l,t} = \prod_{(v,u)\in\mathcal{U}} \mathrm{Bern}(p_0)$ for fixed $0 < p_0 < 1$; we set $p_0 = 1/2$. The prior is fixed independently of the observed adjacency. Its $p_0 = 1/2$ choice is a maximum-entropy capacity regularizer: $D_{\mathrm{KL}}(\mathrm{Bern}(\phi) \| \mathrm{Bern}(1/2)) = \log 2 - H_{\mathrm{bin}}(\phi)$. Candidate truncation controls sparsity, while this penalty controls the certainty of retained gates. During training, candidate gates use binary Concrete samples from equation 14; noncandidate gates remain zero.

$$\begin{aligned}\mathcal{K}_A^{l,t} = &\sum_{(v,u)\in\mathcal{C}_t} \left[\phi_{vu}^{l,t} \log \frac{\phi_{vu}^{l,t}}{p_0} + (1-\phi_{vu}^{l,t}) \log \frac{1-\phi_{vu}^{l,t}}{1-p_0}\right] \\ &+ (|\mathcal{U}| - |\mathcal{C}_t|) \log \frac{1}{1-p_0}.\end{aligned} \tag{16}$$

Equation 16 is the exact KL of the discrete product distributions. The excluded-edge contribution is computed from counts, and $\mathcal{U}$ is fixed across input records. The constant $|\mathcal{U}| \log 2$ may be omitted from gradient computation while retaining it in the formal KL expression. We optimize this

probability penalty using a differentiable Concrete forward pass. The compression proof applies to the discrete posterior and its full-support prior.

Nondegenerate feature posterior. Aggregate the projected context using normalized sampled weights, keeping an explicit self contribution:

$$\begin{aligned} R_v^{l,t} &= C_v^{l,t} + \frac{\sum_{u \in \mathcal{N}_{k_{\mathrm{nbr}},\Delta t}(v)} \widetilde{M}_{vu}^{l,t} C_u^{l,t}}{\epsilon_a + \sum_{u \in \mathcal{N}_{k_{\mathrm{nbr}},\Delta t}(v)} \widetilde{M}_{vu}^{l,t}}, \\ m_v^{l,t} &= R_v^{l,t} W_m^l + b_m^l, \qquad s_v^{l,t} = \mathrm{softplus}(R_v^{l,t} W_s^l + b_s^l) + \epsilon_s, \\ H_v^{l,t} &= m_v^{l,t} + s_v^{l,t} \odot \xi_v^{l,t}, \qquad \xi_v^{l,t} \sim \mathcal{N}(0, I_{d_h}). \end{aligned} \tag{17}$$

The positive constants ϵ_a and ϵ_s stabilize the denominator and bound the standard deviation away from zero; we use 10^{-8} and 10^{-4}. Here d_h is the hidden width, $\odot$ is componentwise multiplication, $\mathrm{softplus}(x) = \log(1 + e^x)$, and the two affine heads produce the mean m and standard deviation s. Conditional on the available history, the feature posterior is a product of $\mathcal{N}(m_v^{l,t}, \mathrm{diag}((s_v^{l,t})^2))$; the prior is a product of standard Gaussians. Their KL is

$$\mathcal{K}_X^{l,t} = \frac{1}{2} \sum_{v,j} \left[(m_{vj}^{l,t})^2 + (s_{vj}^{l,t})^2 - 1 - \log(s_{vj}^{l,t})^2 \right], \qquad j = 1, \ldots, d_h. \tag{18}$$

Equation 18 follows by inserting the Gaussian log densities and using $\mathbb{E}(H - m)^2 = s^2$ and $\mathbb{E}H^2 = m^2 + s^2$ coordinatewise. The positive scale ensures absolute continuity with respect to the Gaussian prior. The stochastic feature head therefore defines the finite feature KL used in the information-bottleneck bound and its training objective.

Prediction and FMI sampling. The final latent embedding G_e is a deterministic readout of the retained latent states: the last state for node/link prediction, or a causal pooling of the observed trajectory for graph classification. A multilayer perceptron (MLP) with two layers, hidden width 128, ReLU, and a categorical or Bernoulli output supplies the cross-entropy term; the Mooc dot-product link decoder is specified in Appendix F.1. For C_Y classes, use normalized probabilities $(1 - C_Y\epsilon)q_\psi + \epsilon$ with $\epsilon = 10^{-6}$, which bounds log-loss while preserving score ordering. For FMI, X_{obs} contains the observed attribute record associated with a sampled node/window pair and H_e its latent embedding. The critic receives the observed attribute record and its paired latent embedding. This defines the feature-specific information $I(X_{\mathrm{obs}}; H_e)$ under the node/window sampling distribution.

We form joint pairs from the same node/window record and product pairs from independently drawn records under the same marginal sampling distribution. For population inference from a single graph trajectory, temporal dependence is handled under the mixing and uniform-deviation assumptions in Appendix D.2. We use a bilinear critic $f_\theta(x, h) = \bar{x}^\top W_F \bar{h} + b_F$, where $W_F \in \mathbb{R}^{d_x \times d_h}$, $b_F \in \mathbb{R}$, and each input vector is divided by the maximum of its norm and 10^{-8}. The bounded output is $F_\theta = 5 \tanh(f_\theta)$. Each encoder step uses one critic Adam update with learning rate 10^{-4} and 256 joint and 256 independent pairs. The marginal draws are sampled independently with replacement from the available records. Every encoder uses the same critic class, training budget, marginal distribution, and preprocessing, with disjoint critic-training and assessment pairs. The population NWJ functional is a lower bound; its finite-sample estimate remains subject to sampling error (Nguyen et al., 2010; Poole et al., 2019).

Optimization and assessment. We alternate critic ascent on the training-pair NWJ objective and encoder/decoder descent on equation 10, holding the other parameter block fixed. The FMI training regularizer uses sampled feature states. Prediction and cross-model assessment use the latent embedding at inference, which for IDG uses posterior means ($M = \phi$, $H = m$). SMI pairs observed structural records with the corresponding latent embeddings. The metric functional applies to both stochastic training states and frozen inference records under their respective sampling distributions.

Normalization and spectral computation. The training objectives use the raw information sums in equation 10. For descriptive comparison, normalize a raw score $u_{b,d}$ as $(u_{b,d} - u_{\min,d})/(u_{\max,d} - u_{\min,d})$ across models b in dataset d when the denominator is positive. This affine map preserves Pearson correlation and records the score range within the

comparison cohort. Proposition 3.1 concerns task information under its population distribution; cohort-normalized values serve the separate purpose of visual comparison.

Model-agnostic SMI and FMI interface. For a frozen encoder Enc, export final-layer latent embeddings $z_{v,t} = \mathrm{Enc}(G^{1:t})_v \in \mathbb{R}^{d_{\mathrm{enc}}}$, where d_{enc} is that model's output width. Align each vector with the same observed neighborhood $a_{v,t}$ and attribute record $x_{v,t}$. Snapshot models export states after all events in the half-open observation window have been processed; event models export states after processing the shared observed prefix. Shared record draws use node identities and cutoffs available to every compared model. SMI receives $(a_{v,t}, z_{v,t})$, and FMI receives $(x_{v,t}, z_{v,t})$. Both scores are computed directly from these pairs, so a latent embedding matrix is a complete latent-side input.

SMI uses final-layer vectors from every model and the same structural descriptors, Gaussian kernels, and record budget. All Gram matrices have size $b \times b$, independent of latent embedding width. For FMI, the critic matrix has shape $d_x \times d_{\mathrm{enc}}$, with the same bounded bilinear architecture and input normalization. We fit a fresh critic for 1,000 Adam updates on training-split records with 256 joint and 256 independent pairs per update, then average 20 held-out assessment batches. Encoders remain frozen during this assessment. The same optimizer, update count, record draws, and attribute preprocessing are used across models; only the critic's input width follows the exported latent embedding width.

Conditional on a finite record pool, deterministic and stochastic encoders both yield well-defined paired-record laws. For deterministic outputs, the empirical joint law is discrete even when the stored vectors have real-valued coordinates: every positive joint mass has positive marginal masses, giving $P \ll Q$ and $I(X_{\mathrm{obs}}; H_e) \leq \log N_{\mathrm{rec}}$. If all paired inputs and outputs are distinct under uniform sampling, empirical MI equals $\log N_{\mathrm{rec}}$ for every such encoder. The bounded-critic FMI score measures accessible dependence at its chosen resolution; latent embedding geometry and critic approximation affect its value. Its population NWJ functional lower-bounds information under the assumptions in Section 3.2.2, while finite-sample estimates retain sampling error.

The compression bound in Appendix E.1 concerns the random latent trajectory. Mean substitution defines a deterministic inference encoder with its own input and representation law. The task-risk identity applies separately to that encoder when the stated task Markov condition holds; the stochastic compression analysis uses the sampled trajectory, and inference assessment uses the paired-record law induced by its deterministic output.

B Related work

B.1 Dynamic graph representation learning

Dynamic graph neural networks (DyGNNs) generally combine graph aggregation with a temporal module, but their input assumptions and prediction interfaces differ substantially. Snapshot-based methods process an ordered sequence of graphs: DySAT alternates structural and temporal attention (Sankar et al., 2020), EvolveGCN recurrently evolves graph-convolution parameters (Pareja et al., 2020), and recent models use spectral or state-space constructions. In particular, SLATE builds a supra-Laplacian encoding across snapshots (Karmim et al., 2024), while GraphSSM derives a Laplacian-regularized state-space model for long graph sequences (Li et al., 2024). Temp-G^{3}NTK instead develops an analyzable temporal graph kernel with graphon convergence (Tieu et al., 2024). Dy-SIGN uses spiking dynamics for time-indexed node classification (Yin et al., 2024), and DG-Mamba scans snapshots with selective state-space mechanisms (Yuan et al., 2025). The 2026 models DyCIL and LAMP add causal invariance and adaptive memory retention, respectively (Zhang et al., 2026; De Castelli et al., 2026). These methods improve the encoder or kernel used to obtain latent embeddings; they do not directly provide two complementary, label-free measures of how much evolving structure and feature dependence the resulting latent embedding retains.

Event-driven methods operate on timestamped interactions rather than pre-aggregated snapshots. DyRep uses temporal point processes (Trivedi et al., 2019), Jodie projects dynamic user/item latent embeddings (Kumar et al., 2019), TGN maintains event-updated memories (Rossi et al., 2020), and PINT develops injective temporal message passing (Souza et al., 2022). TPNet couples structural connectivity and time decay through temporal walk matrices (Lu et al., 2024). RAG4DyG retrieves

contextually and temporally related histories before prediction (Wu et al., 2025). IDG converts event streams to a fixed snapshot protocol and provides intrinsic assessment and optimization of the latent embedding. Consequently, comparisons require a shared chronological split, negative sampler, decoder, and temporal window to assess latent embedding quality under clearly specified prediction and observation interfaces for each temporal encoder.

B.2 Intrinsic representation assessment

Temporal graph assessment also distinguishes prediction behavior from information retained in a latent embedding. Volatility cluster statistics measure whether prediction errors concentrate in nearby time intervals and can guide temporal training objectives (Su & Wu, 2025). This supervised temporal diagnostic complements the label-free input dependence measured by SMI and FMI. For graph-level learning, T3former combines sliding-window topological and spectral descriptors through descriptor attention (Uddin et al., 2026). Its use of spectral structure provides a related motivation for examining the information encoded by structural operators, while SMI evaluates paired input and latent embedding operators directly.

Most DyGNN studies select models through downstream AUC, AP, or classification accuracy. General representation assessment offers label-free alternatives. Alignment and uniformity characterize positive-pair concentration and global dispersion (Wang & Isola, 2020); DGI maximizes dependence between local patches and a graph summary (Veličković et al., 2019); MADGap measures graph over-smoothing (Chen et al., 2020); and RankMe uses the entropy of singular values as an effective-rank diagnostic (Garrido et al., 2023). More recent work uses metric-space magnitude to quantify multi-scale diversity in latent embeddings (Limbeck et al., 2024), while analysis of graph contrastive learning shows that stronger alignment need not monotonically improve generalization (Liu et al., 2024). These measures capture geometry, smoothness, dimensionality, or static dependence, whereas a latent embedding of a dynamic graph must also preserve changes in topology and node attributes across time.

IDG separates these two sources. SMI computes input and latent embedding mutual information from joint and marginal kernel spectra in Section 3.2.1; FMI estimates attribute and latent embedding dependence through an NWJ critic. The resulting scores have explicit paired-record distributions and finite-sample interpretation. SMI and FMI thus provide complementary input and latent embedding dependence coordinates alongside geometric diagnostics and task scores. The perturbation, critic-capacity, and distribution-shift studies in Section G test this separation without claiming that either metric alone is sufficient for every downstream task.

B.3 Information bottlenecks and distribution shift

The information bottleneck compresses an input while preserving task-relevant information (Tishby et al., 2000; Alemi et al., 2017). Graph Information Bottleneck extends this principle to graph structure and attributes (Wu et al., 2020); Dynamic Graph Information Bottleneck further directs structural and feature information across snapshots for robust latent embeddings (Yuan et al., 2024). IDG differs in purpose and decomposition: its bottleneck remains task-aware, while SMI and FMI explicitly diagnose the structural-evolution and feature-dependence information that survives compression. Proposition 3.1 separates target uncertainty, discarded task information, and readout mismatch. It complements the empirical metric correlations by identifying the task-specific information needed for prediction.

Distribution-shift methods directly optimize robustness. DIDA studies spatio-temporal shifts in dynamic graphs (Zhang et al., 2022); EPoD infers evolving environments and mediating dynamic subgraphs (Yang et al., 2024); and GraphMETRO combines aligned experts for heterogeneous graph shifts (Wu et al., 2024). These objectives complement rather than duplicate IDG: robustness methods change the predictor to handle a shift, whereas IDG measures which structural or feature information changed in a frozen latent embedding. The DIDA experiment in Appendix G.8 accordingly reports both attribution and post-shift correlations, under the specified distribution shifts.

Table 4: Positioning of representative related directions. "Intrinsic" denotes a label-free diagnostic applied to the latent embedding rather than a downstream task score.

Direction	Temporal input	Explicit structure	Explicit features	Intrinsic	Main objective
Snapshot DyGNNs	Snapshots	Yes	Yes	No	Prediction
Event DyGNNs	Events	Yes	Optional	No	Prediction
OOD graph learning	Either	Shift-specific	Shift-specific	No	Robust prediction
Static diagnostics	Static	Metric-specific	Metric-specific	Yes	Diagnosis
Self-supervised graph learning	Static	Yes	Optional	Training signal	Representation learning
Dynamic graph IB	Snapshots	Yes	Yes	No	Compression/robustness
IDG	Snapshots	Yes	Yes	SMI/FMI	Diagnosis + learning

B.4 Benchmarks and evaluation protocols

Task-specific benchmarking further clarifies what an evaluation measures. BAG compares dynamic graph anomaly detectors across node and edge tasks, distinguishing synthetic from naturally occurring anomalies (Hua et al., 2026). This separation motivates reporting the task, label construction, and evaluation population together with the metric. Our link prediction, node classification, node-affinity ranking, and graph classification results therefore retain their respective protocols and score definitions, while SMI and FMI provide a common latent embedding assessment interface across all eleven trained models using the same paired-record construction.

Recent benchmarks emphasize that temporal-graph conclusions are sensitive to scale and protocol. TGB 2.0 provides large temporal knowledge and heterogeneous graphs with reproducible future-link evaluation (Gastinger et al., 2024), while DTGB jointly varies graph structure and textual attributes (Zhang et al., 2024). They also illustrate why single-relation snapshot graphs, heterogeneous temporal graphs, and temporal knowledge graphs should not be treated as identical input classes. In this work, every event stream is chronologically discretized before IDG is applied, and controlled studies keep the raw events, split, training budget, and negative sampling fixed. This design isolates the assessment signal from window selection and readout design as far as the recorded protocols permit.

C Derivation of latent embedding likelihood

Equation 2 provides the generative motivation for a structured latent state. We derive its edge-agreement score below; the implemented training objective is equation 10. The goal of the latent embedding is to summarize the holistic evolutionary information of a dynamic graph sequence in a compact graph-like object. Consider a dynamic graph sequence $\mathcal{G} = \{G_1, \ldots, G_T\}$, where the i-th snapshot is $G^i = (V^i, E^i, X^i, A^i)$. For this structural derivation, the latent embedding is $G_e = (V_{G_e}, E_{G_e}, M, H)$, with H held fixed, where

$$M_{\alpha\beta} = \begin{cases} 1, & (\alpha, \beta) \in E_{G_e}, \\ 0, & \text{otherwise}, \end{cases} \qquad A^i_{ab} = \begin{cases} 1, & (a, b) \in E^i, \\ 0, & \text{otherwise}. \end{cases} \tag{19}$$

For each snapshot, the assignment matrix S^i specifies the correspondence from snapshot nodes to latent nodes:

$$S^i_{a\alpha} = \begin{cases} 1, & F(v^i_a) = \alpha, \\ 0, & \text{otherwise}, \end{cases} \tag{20}$$

where F maps the persistent identity of node a in G^i to its index α on the common latent vertex basis. Let $\mathcal{S}^i_{\text{id}}$ contain the one-hot assignment fixed by these identities. The constrained reconstruction-score formulation is

$$S^i = \arg\max_{S \in \mathcal{S}^i_{\text{id}}} \mathcal{W}(G^i; G_e, S). \tag{21}$$

Given $\sum_\alpha S^i_{a\alpha} = 1$, zero-assignment factors are omitted from the independent edge-noise product:

$$\mathcal{W}(G^i; G_e, S^i) = \prod_{a \in V^i} \prod_{\alpha \in V_{G_e}} \left[\mathcal{W}(A^i_{a\cdot}; G_e, S^i, \alpha)\right]^{S^i_{a\alpha}}. \tag{22}$$

We parameterize the node-to-latent prior as $B^i_{a\alpha} \geq 0$ with $\sum_\alpha B^i_{a\alpha} = 1$, and set

$$\mathcal{W}(A^i_{a\cdot}; G_e, S^i, \alpha) = B^i_{a\alpha} \prod_{b \in V^i} P(A^i_{ab} \mid G_e, S^i_{b\cdot}, \alpha), \tag{23}$$

The identity constraint fixes the assignment without a separate matching step. We set the normalized prior to $B^i_{a\alpha} = S^i_{a\alpha}$. The factors define an assignment-prior-weighted edge-agreement score $\mathcal{W}$ under the specified observation model. Its product form specifies edge agreement under the fixed alignment used by the causal encoder.

For the mapped latent neighbor state, define

$$\widehat{A}^i_{ab|\alpha} = \sum_{\beta \in V_{G_e}} M_{\alpha\beta} S^i_{b\beta}, \qquad c^i_{ab\alpha} = A^i_{ab}\widehat{A}^i_{ab|\alpha} + (1 - A^i_{ab})(1 - \widehat{A}^i_{ab|\alpha}). \tag{24}$$

The one-hot condition makes $\widehat{A}^i_{ab|\alpha}$ binary. The indicator $c^i_{ab\alpha}$ equals one for agreement on either an edge or a non-edge, and equals zero for disagreement. With edge-error probability P_e, the symmetric observation model is

$$P(A^i_{ab} \mid G_e, S^i_{b\cdot}, \alpha) = (1 - P_e)^{c^i_{ab\alpha}} P_e^{1 - c^i_{ab\alpha}}. \tag{25}$$

Substituting equation 25 into equation 23 yields

$$\begin{aligned}
\mathcal{W}(A^i_{a\cdot}; G_e, S^i, \alpha) &= B^i_{a\alpha} \prod_{b \in V^i} (1 - P_e)^{c^i_{ab\alpha}} P_e^{1 - c^i_{ab\alpha}} \\
&= B^i_{a\alpha} P_e^{|V^i|} \left(\frac{1 - P_e}{P_e} \right)^{\sum_{b \in V^i} c^i_{ab\alpha}} \\
&= K^i_{a\alpha} \exp\left(\mu \sum_{b \in V^i} c^i_{ab\alpha} \right),
\end{aligned} \tag{26}$$

where $\mu = \ln \frac{1 - P_e}{P_e}$ and $K^i_{a\alpha} = P_e^{|V^i|} B^i_{a\alpha}$. Finally, substituting equation 26 into equation 22 gives the score for one snapshot:

$$\mathcal{W}(G^i; G_e, S^i) = \prod_{a \in V^i} \prod_{\alpha \in V_{G_e}} \left[K^i_{a\alpha} \exp\left(\mu \sum_{b \in V^i} c^i_{ab\alpha} \right) \right]^{S^i_{a\alpha}}. \tag{27}$$

Multiplying the snapshot scores over the sequence $\mathcal{S} = \{S^1, \ldots, S^T\}$ gives

$$\mathcal{W}(\mathcal{G}; G_e, \mathcal{S}) = \prod_{G^i \in \mathcal{G}} \mathcal{W}(G^i; G_e, S^i), \tag{28}$$

and equation 2 follows immediately by substituting equation 27 into equation 28.

D Derivation of SMI and FMI

D.1 Input and latent embedding structure mutual information

This subsection gives the complete probability construction for SMI and its sparse implementation. The two measured objects are an input structural record and the corresponding latent embedding. A node/window record supplies their common observational unit, so the pairing remains defined for a single snapshot as well as for an evolving sequence.

Sparse structural descriptor. Let $\widetilde{A}_t = (A_t + A_t^\top)/2$ be the nonnegative observed adjacency on persistent node identities, with a zero diagonal. Multiple interactions within a window are accumulated as edge weights. Write $d^t_v = \sum_u \widetilde{A}_{t,vu}$ and define

$$a_{v,t} = \left[\frac{\widetilde{A}_{t,v:}}{\max(1, d^t_v)} \,\middle\|\, \log(1 + d^t_v) \right], \qquad z^l_{v,t} = \mathrm{Enc}^l(G^{1:t})_v. \tag{29}$$

Here Enc^l exports a model's node latent embedding at the observation cutoff; for IDG, $z^l_{v,t} = m^{l,t}_v$. The sparse neighborhood profile retains adjacency information, and the final coordinate retains degree magnitude. All profiles use the same vertex coordinates; missing entries are zero. The descriptor uses the observed graph before candidate-neighborhood truncation, covering every observed neighbor. This structural diagnostic measures symmetrized topology. Only vertices and edges observed by the prediction cutoff enter the descriptor.

For paired records indexed by i, form

$$k_A(a_i, a_j) = \exp\left(-\frac{\|a_i - a_j\|_2^2}{2h_A^2}\right), \qquad k_Z(z_i, z_j) = \exp\left(-\frac{\|\bar{z}_i - \bar{z}_j\|_2^2}{2h_Z^2}\right),$$

where $h_A, h_Z > 0$ are bandwidths and $\bar{z}_i = z_i / \max(\|z_i\|_2, 10^{-8})$. We use $h_Z = 1$ and set h_A to the median positive pairwise distance of training structural descriptors, with value 1 for a degenerate training sample. This input bandwidth is then fixed across models, layers, and splits. The node/window distribution first selects an observed window uniformly and then an active node uniformly. The same sampled records are used for every model in a comparison.

Joint state and marginals. Positive-definite Gaussian kernels admit unit-norm feature maps $\varphi(a)$ and $\psi(z)$ in reproducing-kernel Hilbert spaces (Sanchez Giraldo et al., 2015; Yu et al., 2021). For an arbitrary joint law of (A_{obs}, Z_e), define

$$\begin{aligned} \Sigma_{AZ} &= \mathbb{E}\big[|\varphi(A_{\mathrm{obs}})\rangle\langle\varphi(A_{\mathrm{obs}})| \otimes |\psi(Z_e)\rangle\langle\psi(Z_e)|\big], \\ \Sigma_A &= \mathrm{Tr}_Z\, \Sigma_{AZ}, \qquad \Sigma_Z = \mathrm{Tr}_A\, \Sigma_{AZ}, \\ I_{\mathrm{sp}}(A_{\mathrm{obs}}; Z_e) &= D_{\mathrm{q}}(\Sigma_{AZ} \| \Sigma_A \otimes \Sigma_Z). \end{aligned} \tag{30}$$

The partial trace Tr_Z integrates out the latent feature space, and Tr_A integrates out the structural feature space. Unit norms make the joint operator positive semidefinite with trace one and give the stated marginals. For the finite empirical law assigning mass $1/b$ to each paired record, all operators have finite rank, so their spectral entropies are finite without an assumption on the differential entropy of raw features.

Proof of the Gram formula. Place the vectors $b^{-1/2}\varphi_i$ in the columns of a feature operator V_A. Then $V_A V_A^* = \Sigma_A$ and $V_A^* V_A = B_A$, which have identical nonzero eigenvalues. The same argument applies to V_Z. For the joint operator, use columns $b^{-1/2}(\varphi_i \otimes \psi_i)$. Their inner products equal $b^{-1}k_A(a_i, a_j)k_Z(z_i, z_j)$, exactly the entries of $B_{AZ} = b(B_A \circ B_Z)$. Thus the joint and marginal spectral entropies in equation 3 are those of one consistent bipartite state, and expanding its relative entropy gives equation 4.

Nonnegativity and upper bound. Quantum relative entropy is nonnegative and vanishes exactly when $\Sigma_{AZ} = \Sigma_A \otimes \Sigma_Z$ (Watrous, 2018). Moreover, $B_{AZ} = (bB_Z) \circ B_A$ is the image of B_A under a trace-preserving unital Schur channel, because bB_Z is positive semidefinite with unit diagonal. Such a channel cannot decrease von Neumann entropy, giving $S(B_{AZ}) \geq S(B_A)$. Interchanging the two kernels gives $S(B_{AZ}) \geq S(B_Z)$. Combining these inequalities with nonnegativity yields the upper bound in Section 3.2.1. Independence of the population input and latent variables factors the joint state and gives zero structure mutual information. A finite paired sample can retain estimation bias, so finite-sample scores are compared under the same sampling budget.

Structural fidelity and practical scope. If all latent embeddings coincide, $B_Z = \mathbf{1}\mathbf{1}^\top/b$, $B_{AZ} = B_A$, and SMI is zero. If the latent kernel reproduces a nonconstant structural kernel, the two sides retain common structural variation even when the graph is stationary in time. A simultaneous permutation of the observed nodes permutes descriptor coordinates and paired records, preserving all pairwise distances and spectra. Orthogonal rotation of latent embeddings also preserves the score. SMI therefore assesses information retained in the input and latent embedding pairing. The descriptor, input normalization, and kernel bandwidth define the shared resolution used for comparing structural fidelity across different encoder representations.

Computation and numerical precision. We use $b = \min(128, N_{\mathrm{rec}})$ paired records per objective block, where N_{rec} is the available node/window record count. We form the sparse-row distance

matrix from row norms and sparse inner products, then evaluate three $b \times b$ Gram spectra for each layer. Assessment averages 20 batches with fixed shared record draws and uses the final layer of every model. We compute eigenvalues in float64, replace round-off negative eigenvalues by zero, renormalize their sum to one, and evaluate $-\nu \log \nu$ only at positive eigenvalues. Automatic differentiation uses a 10^{-12} floor inside the logarithm. The finite precision convention stabilizes gradients while the displayed identities describe the exact spectral functional.

Resolution and interpretation. The record budget controls sampling resolution, and the bandwidth controls which neighborhood differences receive similar kernel values. Both are held fixed during cross-model comparison. This construction evaluates sparse observed neighborhoods directly and uses all coordinates through their sparse inner products, including coordinates associated with newly observed neighbors. Population dependence, finite-sample structure mutual information, and downstream prediction are evaluated at their respective observational levels; the correlation experiment measures the empirical relationship between the latter two.

D.2 Derivation and consistency of FMI estimation

The FMI between the latent embedding and its observed attribute record is the KL divergence between the joint distribution and the product of its marginals. We write $P = \mathbb{P}_{X_{\text{obs}} H_e}$ and $Q = \mathbb{P}_{X_{\text{obs}}} \otimes \mathbb{P}_{H_e}$.

$$\text{FMI}(G, G_e) = D_{\text{KL}}(P \| Q) = \sup_{F : \mathcal{X} \times \mathcal{H} \to \mathbb{R}} \left\{ \mathbb{E}_P[F] - \mathbb{E}_Q[e^{F-1}] \right\}, \tag{31}$$

where $\mathcal{X}$ and $\mathcal{H}$ are the attribute and latent-feature spaces. The optimizer is $F^\star = 1 + \log(dP/dQ)$ when finite and admissible; zero density ratios can require a limiting sequence. Restricting the supremum to $\mathcal{F}_\Theta$ gives only a lower bound. For finite KL, the identity follows from the variational inequality and its attainable or limiting equality case:

$$D_{\text{KL}}(P \parallel Q) \geq \mathbb{E}_P[F] - \mathbb{E}_Q\left[e^{F-1}\right] = \sum_i p_i f_i - \sum_i q_i e^{f_i - 1}. \tag{32}$$

In the discrete illustration below, p_i, q_i are joint and product-marginal masses on atom i, and $f_i = F(i)$; continuous distributions use the density-ratio argument in the main text. To find the extremum of equation 32, we take the derivative with respect to f_j and set it equal to zero. For readability, we abbreviate the original distributions by writing P for $\mathbb{P}_{X_{\text{obs}} H_e}$ and Q for $\mathbb{P}_{X_{\text{obs}}} \otimes \mathbb{P}_{H_e}$ throughout the optimization steps below in this derivation:

$$\begin{aligned} &\frac{\partial \left[\sum_i p_i f_i - \sum_i q_i e^{f_i - 1}\right]}{\partial f_j} = 0 \\ &\implies p_j - q_j e^{f_j - 1} = 0 \\ &\implies e^{f_j - 1} = \frac{p_j}{q_j} \\ &\implies f_j = \log \frac{p_j}{q_j} + 1. \end{aligned} \tag{33}$$

Substitute f_j into f_i in the expression $\sum_i p_i f_i - \sum_i q_i e^{f_i - 1}$:

$$\begin{aligned} \sum_i p_i f_i - \sum_i q_i e^{f_i - 1} &= \sum_i p_i \left(\log \frac{p_i}{q_i} + 1 \right) - \sum_i q_i e^{f_i - 1} \\ &= \sum_i p_i \log \frac{p_i}{q_i} + \sum_i p_i \cdot 1 - \sum_i q_i \cdot \frac{p_i}{q_i} \\ &= \sum_i p_i \log \frac{p_i}{q_i} + 1 - 1 \\ &= \sum_i p_i \log \frac{p_i}{q_i} = D_{\text{KL}}(P \parallel Q). \end{aligned} \tag{34}$$

For positive masses the bound is tight at $F^\star$; atoms with $p_i = 0 < q_i$ use $f_i \to -\infty$ and $0 \log 0 = 0$. In practice, let $(x_i, h_i) \sim P$ and $(\widetilde{x}_i, \widetilde{h}_i) \sim Q$ be paired positive and independently paired negative samples. We use

$$\widehat{\mathrm{FMI}}_n = \sup_{\theta \in \Theta} \left[\frac{1}{n} \sum_{i=1}^{n} F_\theta(x_i, h_i) - \frac{1}{n} \sum_{i=1}^{n} e^{F_\theta(\widetilde{x}_i, \widetilde{h}_i) - 1} \right]. \tag{35}$$

Write $J_\theta = \mathbb{E}_P F_\theta - \mathbb{E}_Q e^{F_\theta - 1}$ and let $\widehat{J}_\theta$ be its empirical counterpart. On an event where $\sup_\theta |J_\theta - \widehat{J}_\theta| \le \varepsilon_{\mathrm{est}}$, an $\varepsilon_{\mathrm{opt}}$-optimal empirical critic $\widehat{\theta}$ satisfies

$$\begin{aligned} I(X_{\mathrm{obs}}; H_e) - J_{\widehat{\theta}} &= \varepsilon_{\mathrm{app}} + \sup_\theta J_\theta - J_{\widehat{\theta}} \\ &\le \varepsilon_{\mathrm{app}} + 2\varepsilon_{\mathrm{est}} + \varepsilon_{\mathrm{opt}}, \end{aligned}$$

where $\varepsilon_{\mathrm{app}} = I(X_{\mathrm{obs}}; H_e) - \sup_\theta J_\theta$. For the empirical statistic itself, the same event gives $|I(X_{\mathrm{obs}}; H_e) - \widehat{J}_{\widehat{\theta}}| \le \varepsilon_{\mathrm{app}} + \varepsilon_{\mathrm{est}} + \varepsilon_{\mathrm{opt}}$. These distinguish error in the population objective of the fitted critic from error in its observed numerical estimate.

For a finite class of K_F critics bounded in $[-M_F, M_F]$ and n i.i.d. pairs from each of P and Q, Hoeffding's inequality and a union bound over both expectations give, with probability $1 - \delta$,

$$\varepsilon_{\mathrm{est}} \le \left(2M_F + e^{M_F - 1} - e^{-M_F - 1}\right) \sqrt{\frac{\log(4K_F/\delta)}{2n}}.$$

For an infinite neural class, a covering-number or empirical-process bound replaces the finite-class term K_F. Temporally dependent samples require a uniform deviation bound under an explicit mixing law or independent-block construction. These extensions connect the capacity term and sampling correction to the selected critic and record distribution. Consistency follows when the approximation error of a growing critic class, its uniform sampling deviation, and the optimization error all tend to zero as the data and optimization budgets increase.

E Derivation of the bottleneck objective

This appendix unpacks the compression and prediction sides of the bottleneck objective separately, so that the variational approximations used in the main text can be followed step by step without skipping intermediate dependencies in the derivation chain.

E.1 Compression term

The compression term measures source information retained by the random latent embedding:

$$I(\mathcal{G}; G_e) = \mathbb{E} \log \frac{P(G_e \mid \mathcal{G})}{P(G_e)}. \tag{36}$$

We condition throughout on fixed encoder parameters, the vertex schema, and the prediction query. Initial states are fixed or independent of the input. Each prior is input-independent and covers the posterior support. These conditions define a common probability space and allow dependencies between observations at arbitrary temporal distances.

Causal update order. At layer l and window t, write $G_{e_A}^{l,t}$ for the sampled structure and $G_{e_X}^{l,t}$ for the sampled feature state. Order updates by time, then layer, with structure before features. Their preceding random histories are

$$\begin{aligned} H_A^{l,t} &= \{G_{e_A}^{k,s}, G_{e_X}^{k,s} : s < t\} \cup \{G_{e_A}^{k,t}, G_{e_X}^{k,t} : k < l\}, \\ H_X^{l,t} &= H_A^{l,t} \cup \{G_{e_A}^{l,t}\}. \end{aligned} \tag{37}$$

Sets contain only updates present in the encoder. The structural posterior conditions on $(G^{1:t}, H_A^{l,t})$ and the feature posterior on $(G^{1:t}, H_X^{l,t})$. Their product is the conditional joint distribution at that layer and window. The full observed prefix includes the raw neighbor cache and timestamps used in candidate selection, so their dependence on the input remains accounted for.

Lemma E.1. *For a Markov chain $X \to Z \to Y$, data processing gives $I(X;Y) \le I(X;Z)$ (Cover & Thomas, 2006).*

Let $Z_{1:J}$ list all random updates in this order, with J their total count. The readout G_e uses only these states, ensuring $\mathcal{G} \to Z_{1:J} \to G_e$. Then

$$I(\mathcal{G}; G_e) \le I(\mathcal{G}; Z_{1:J}) = \sum_{j=1}^{J} I(\mathcal{G}; Z_j \mid Z_{<j}). \tag{38}$$

A readout of the final state or pooling of retained states meets this condition. An additional raw-input path must be included in the information accounting. Node identities and prediction queries are fixed interfaces under the conditioning specified above.

Conditional variational identity. For an update Z_j at time $t(j)$, causality gives $P(Z_j \mid \mathcal{G}, Z_{<j}) = P(Z_j \mid G^{1:t(j)}, Z_{<j})$. Adding and subtracting the conditional marginal log-density yields

$$\begin{aligned} I(\mathcal{G}; Z_j \mid Z_{<j}) = \mathbb{E} D_{\mathrm{KL}}\Big(P(Z_j \mid G^{1:t(j)}, Z_{<j}) \| Q_j\Big) \\ - \mathbb{E} D_{\mathrm{KL}}(P(Z_j \mid Z_{<j}) \| Q_j)\,. \end{aligned} \tag{39}$$

The second term is nonnegative. Summing over the layers at each window therefore gives

$$\sum_l I(\mathcal{G}; G_{e_A}^{l,t} \mid H_A^{l,t}) \le \sum_l \mathbb{E} D_{\mathrm{KL}}\Big(P(G_{e_A}^{l,t} \mid G^{1:t}, H_A^{l,t}) \| Q_A^{l,t}\Big) = \mathrm{AGIB}^{(t)}, \tag{40}$$

$$\sum_l I(\mathcal{G}; G_{e_X}^{l,t} \mid H_X^{l,t}) \le \sum_l \mathbb{E} D_{\mathrm{KL}}\Big(P(G_{e_X}^{l,t} \mid G^{1:t}, H_X^{l,t}) \| Q_X^{l,t}\Big) = \mathrm{XGIB}^{(t)}\,. \tag{41}$$

Combining these bounds with equation 38,

$$I(\mathcal{G}; G_e) \le \sum_{t \in S_A} \mathrm{AGIB}^{(t)} + \sum_{t \in S_X} \mathrm{XGIB}^{(t)}\,. \tag{42}$$

The exact Bernoulli and nondegenerate Gaussian KL terms are given in equations 16 and 18. Their expectations are under the stochastic encoder trajectory, with all layer and time updates included. Finite KL suffices for these identities. Estimating their expectations from one dependent trajectory additionally requires control of sampling fluctuations.

Scope of the bound. The compression guarantee applies to the discrete stochastic encoder. binary Concrete draws provide differentiable training surrogates, while posterior-mean substitution defines deterministic inference. The discrete compression bound and the deterministic inference law are analyzed separately. Proposition 3.1 applies separately to any inference encoder satisfying its task Markov condition, including the mean encoder.

E.2 Predictive term

For a finite-valued target Y and a normalized predictor $q_\psi(Y \mid G_e)$, expand the mutual information and add the predictive log-likelihood:

$$\begin{aligned} I(Y; G_e) &= H(Y) + \mathbb{E} \log P(Y \mid G_e) \\ &= H(Y) + \mathbb{E} \log q_\psi(Y \mid G_e) + \mathbb{E}_{G_e} D_{\mathrm{KL}}(P(Y \mid G_e) \| q_\psi(Y \mid G_e)) \\ &\ge H(Y) + \mathbb{E} \log q_\psi(Y \mid G_e). \end{aligned} \tag{43}$$

For supervised node labels $Y = (Y_v)_{v \in V_{\mathrm{sup}}}$, a product decoder $q_\psi(Y \mid G_e) = \prod_{v \in V_{\mathrm{sup}}} q_\psi(Y_v \mid H_v)$ is normalized even when the true labels are dependent. It yields

$$I(Y; G_e) \ge H(Y) - \mathbb{E} \sum_{v \in V_{\mathrm{sup}}} \mathrm{CE}(q_\psi(\cdot \mid H_v), Y_v). \tag{44}$$

Here $\mathrm{CE}(q, y) = -\log q(y)$ is cross-entropy. The label entropy is part of the formal variational bound. It is constant with respect to encoder and decoder parameters, so maximizing the bound is equivalent to minimizing expected cross-entropy. Link and graph tasks use their corresponding supervised edges and graph labels; the predictive lower bound uses this conditional-likelihood construction throughout Algorithm 1.

E.3 INFORMATION RECOVERY AND THE REGULARIZED OBJECTIVE

The following result links the two assessment objectives to recoverable input information. It uses the final latent embedding Z, the structural record $A_{\mathrm{obs}} = a(A)$, and the feature record $X_{\mathrm{obs}} = x(X)$. The maps a and x are fixed preprocessing maps. All quantities in an inequality refer to the same paired-record probability law and the same encoder.

Proposition E.2 (Input recovery and task log-risk)**.** *Let $D = (A, X)$ have finite support, let Y be a finite target, and let $Y \to D \to Z$. Let I_{sp} be the population structure mutual information of (A_{obs}, Z) with finite spectral entropies, and let $J_F = \mathbb{E}_P F - \mathbb{E}_Q e^{F-1}$ be the population NWJ functional for (X_{obs}, Z) of a critic for which both expectations are finite. Then*

$$I_{\mathrm{sp}}(A_{\mathrm{obs}}; Z) \le I(A_{\mathrm{obs}}; Z) \le I(A; Z), \tag{45}$$

$$H(A \mid Z) + H(X \mid Z) \le H(A) + H(X) - I_{\mathrm{sp}}(A_{\mathrm{obs}}; Z) - J_F, \tag{46}$$

$$\mathcal{R}^*_{\log}(Z) \le H(Y \mid D) + H(A) + H(X) - I_{\mathrm{sp}}(A_{\mathrm{obs}}; Z) - J_F. \tag{47}$$

The first conditional entropy in equation 46 is the optimal log-loss for reconstructing the structural input from Z; the second is its feature counterpart.

Proof. Prepare the kernel states $|\varphi(a)\rangle\langle\varphi(a)|$ and $|\psi(z)\rangle\langle\psi(z)|$ locally from the two classical records. Monotonicity of relative entropy under these local preparation channels maps the classical joint and product laws to Σ_{AZ} and $\Sigma_A \otimes \Sigma_Z$, respectively (Watrous, 2018). This proves the first inequality in equation 45. Classical data processing gives the second. The NWJ inequality and the fixed feature map give $J_F \le I(X_{\mathrm{obs}}; Z) \le I(X; Z)$. Subtracting these inequalities from $H(A) + H(X)$ proves equation 46. Proposition 3.1 gives $\mathcal{R}^*_{\log}(Z) = H(Y \mid D) + I(Y; D \mid Z)$. Finally, $I(Y; D \mid Z) \le H(D \mid Z) \le H(A \mid Z) + H(X \mid Z)$, which proves equation 47.

Practical conditions and use. Finite support is realized by a fixed observation interval with finite-precision stored attributes and a fixed vertex schema. It allows dependent snapshots, nonlinear encoders, and arbitrary target mechanisms. For deterministic assessment on a finite record pool, the spectral operators also have finite rank. The Markov condition follows by freezing the encoder before evaluation and providing it only the observed input. The theorem applies separately to each latent embedding layer; final-layer assessment uses a single I_{sp} as in the experiments. The training sum in equation 4 applies the structural criterion at each layer.

The input entropies and $H(Y \mid D)$ are constant for a fixed evaluation law. Increasing either dependence term tightens the displayed recovery certificate, while the supervised predictor targets the conditional target distribution. This establishes complementary roles for intrinsic fidelity and task supervision. For a held-out empirical FMI value $\widehat{J}_F$ satisfying $|\widehat{J}_F - J_F| \le \varepsilon$, adding ε to the right-hand side preserves the bound. Thus the population identities describe information recovery, and the sampling corrections quantify estimation uncertainty.

Variational upper bound and slack. Write $K = \sum_t \mathrm{AGIB}^{(t)} + \sum_t \mathrm{XGIB}^{(t)}$ and $U = \beta K - \mathbb{E} \log q_\psi(Y \mid Z)$. For the stochastic encoder in Appendix E.1, equations 42 and 43 give

$$U - H(Y) - \mathcal{L}_{\mathrm{GIB}} = \beta\big[K - I(D; Z)\big] + \mathbb{E}_Z D_{\mathrm{KL}}\big(P(Y \mid Z) \| q_\psi(Y \mid Z)\big) \ge 0. \tag{48}$$

The first term measures compression-bound slack; the second measures readout mismatch. Minimizing U is variational upper-bound optimization. Equation 10 further rewards structural and feature fidelity through empirical SMI and the fitted NWJ objective.

Implementable coefficient conditions. The implemented regularizers have bounded ranges. For L layers and record budget b, $0 \le \mathrm{SMI} \le L \log b$. With $F_\theta = c \tanh f_\theta$ and $c = 5$, the empirical FMI objective is at most $c - e^{-c-1}$. Nonnegative KL and cross-entropy therefore imply, for every $\beta \ge 0$ and $\lambda \ge 0$,

$$\widehat{\mathcal{L}} \ge -\lambda\big(L \log b + c - e^{-c-1}\big). \tag{49}$$

This finite lower bound follows directly from the record budget and critic cap, with the full KL constants retained. Omitting parameter-independent constants shifts the objective and its bound by the same amount. The coefficient β controls stochastic capacity, while λ controls the reward for retained input dependence. Validation selects their balance using the downstream objective. The default pair $(\lambda, \beta) = (0.01, 0.1)$ uses the same bounded estimators as the sensitivity experiments.

F EXPERIMENT SETUP

In comparison tables, **bold** and underline identify the best and second-best distinct reported means within each dataset and metric. Tables 8, 9, 10, and 12 omit rank highlighting for compact comparisons. Tied means share a rank. Scores are maximized and runtime or estimation gaps are minimized. Counts, parameter budgets, signed diagnostic responses, and component costs are descriptive quantities rather than comparative performance scores.

This section records the implementation choices, dataset construction details, baseline protocols, and task definitions used to reproduce the experiments under a consistent dynamic-graph evaluation pipeline throughout the paper and appendix experiments.

F.1 DETAILS FOR IMPLEMENTATION

A multilayer perceptron (MLP) serves as the standard decoder, with a Mooc link-prediction exception: IDG uses a symmetric dot product with random negatives and BCE. Other link-prediction settings use concatenated latent embeddings. In non-TGB link prediction, training uses $N_-^{\text{train}}/N_+^{\text{train}} = 0.1$: 100 negative edges per 1,000 positive edges. Here N_- and N_+ denote negative and positive edge counts. Each test snapshot uses equal positive and negative counts, $N_-^{\text{test}}/N_+^{\text{test}} = 1$; each count is 20% of that snapshot's original edge count. TGB link prediction uses the official benchmark sampler and its prescribed evaluation candidates.

For non-TGB link prediction, positive edges have label 1 and negatives label 0. We compute unweighted positive-class AP from predicted link probabilities using `average_precision_score` with `pos_label=1`. We first average AP over test snapshots, then report its mean and standard deviation across five random seeds. The test ratio, rather than the training ratio, determines the class balance of this ranking evaluation. AUC measures discrimination between positive and negative links. Node classification reports AUC, AP, and recall; graph classification reports ACC, the fraction of correctly classified graph samples.

We use model-specific default encoder configurations with the task protocols specified in this section. Non-TGB link and node prediction use 5 independent runs and a 70:15:15 training/validation/test split; TGB retains official splits. Unless otherwise stated, IDG uses Adam, a 3-layer attention-based causal encoder (Veličković et al., 2018) with hidden width 128, dropout 0.1, and learning rate 10^{-4}. Neural training stops after 100 consecutive epochs without improvement in validation AUC for binary prediction, ACC for graph classification, or NDCG@10 for affinity ranking, with a maximum of 1000 epochs. We restore the best validation checkpoint for test evaluation. The patience is independent of the maximum epoch count. In equation 10, λ weights SMI/FMI. All per-epoch times use one NVIDIA RTX 3090 with 24 GB memory.

Graph-classification resampling. IDG uses 100 bootstrap repetitions with random 80%/10%/10% training/validation/test splits and independent weight initializations, following the graph-level protocol (Micheli & Tortorella, 2022). Model selection uses validation data; means and standard deviations summarize test ACC across repetitions. Table 15 reproduces seven kernel baselines from that paper's Table 4, including APPR-250 as APPR-S. These published results originate from repeated ten-fold evaluation (Oettershagen et al., 2020). We retain their reported deviations and resampling units for a descriptive literature comparison.

F.2 DATASET DETAILS

The datasets selected for our experiments (Table 5) span distinct interaction counts and processing horizons, ranging from short snapshot sequences to large temporal interaction streams. This diversity constitutes a comprehensive foundation for evaluating the model's stability and effectiveness across heterogeneous data scales, graph densities, and application domains.

GitHub dataset. The source is GitHub WatchEvent activity collected from January 1 to January 31, 2015. A WatchEvent records an account starring a repository. The raw collection contains 104,614 vertices and 195,029 interactions. Preprocessing retains 19,664 interactions and constructs 15 chronological snapshots, with an average of 1,844 vertices and 1,214 edges per snapshot. These averages summarize the individual snapshot graphs; retained interactions count event records across

Table 5: Dataset statistics. Raw counts describe the source event collection; retained interactions and windows describe the processed sequence. Graph-classification counts describe sampled temporal graphs.

Link prediction / node classification					
Metrics	Reddit	Mooc	GitHub	Wikipedia	Tgbl-Wiki
Raw vertices	11,000	7,145	104,614	9,227	9,227
Raw interactions	672,447	411,749	195,029	157,474	157,474
Retained interactions	10,000	411,749	19,664	10,000	157,474
Processed windows	20	20	15	20	152,757

Graph classification						
Metrics	Dblp	Facebook	Highschool	Infectious	Tumblr	MIT
Nodes (avg.)	53	96	52	50	53	20
Edges (avg.)	320	269	545	459	199	1,469
Sample graphs	755	995	180	200	373	97

the full processed sequence. Our account-level preprocessing represents users or organizations and derives outgoing and incoming event-role features. We accumulate node features within each window. The binary node task distinguishes individual users from organization accounts within this preprocessing protocol for binary node classification.

Mooc dataset (Kumar et al., 2019): A directed temporal network representing learner actions on a Mooc platform. It contains 411,749 interactions between 7,047 users and 98 course activities. Edges represent user actions with timestamps, labeled as 1 (dropout/exit) or 0. User features are derived by summing edge features, while target node features are computed via edge feature differences. Because the original benchmark is an event stream, we retain all interaction records with valid timestamps, divide the total time span into fixed intervals, and construct 20 consecutive snapshots. The task is to predict user course exit in the next stage of the interaction sequence during temporal dropout assessment within the same benchmarking pipeline used here.

Wikipedia dataset (Kumar et al., 2019): This dataset derives from one month of editing data on the 1,000 most active pages. The original set includes 8,227 active users ($\geq$ 5 edits) and 157,474 interactions. Edit texts were converted into semantic vectors using Linguistic Inquiry and Word Count (LIWC). Labels denote historical user bans (217 positive). Since the benchmark is released as timestamped edit events, we retain the first 10,000 interactions for scalability, order them by time, and partition them into 20 consecutive snapshots for binary node classification.

Reddit dataset (Kumar et al., 2019): A temporal user/subreddit interaction network recording users' posts to subreddits. Each interaction includes a timestamp and LIWC text features; it is distinct from the subreddit-to-subreddit hyperlink dataset. The original one-month sample covers 1,000 active subreddits, 10,000 users, and 672,447 interactions, with post texts mapped to LIWC feature vectors. Labels are derived from ban records (366 labeled nodes). As with Wikipedia, we treat the original interaction log as a timestamped event stream, retain the first 10,000 interactions in chronological order, and divide the time span into fixed intervals to construct 20 consecutive snapshots. The task is binary node classification.

Tgbl-Wiki dataset (Huang et al., 2023): This dataset stores the temporal co-editing network on Wikipedia pages over a one month period. The network is structured as a dynamic bipartite interaction graph where both editors and wiki pages function as nodes. An edge represents a specific user editing a page at a recorded timestamp. Crucially, each edge is associated with rich text features directly derived from the content of the page edits. The original TGB benchmark is a timestamped event stream with 152,757 nonempty 1-second snapshots. To evaluate IDG in the snapshot-based setting, we sort all events by timestamp and assemble causal windows using the protocol in Appendix G.9. We utilize this dataset primarily for the link prediction task to evaluate the model's ability to accurately forecast future user-page interactions.

Graph classification datasets To evaluate the model's capability in global property prediction, we employ six dynamic graph datasets derived from real-world dynamic social interaction networks: **Dblp**, **Facebook**, **Highschool**, **Infectious**, **MIT**, and **Tumblr**. Following the experimental protocols established in prior work (temporal graph kernels (Oettershagen et al., 2020)), these datasets serve as the topological foundation for constructing synthetic dissemination processes. The graph-classification benchmarks contain temporal graph samples: 755 for Dblp, 995 for Facebook, 180 for Highschool, 200 for Infectious, 373 for Tumblr, and 97 for MIT. These counts in Table 5 are numbers of sample graphs, not temporal steps within one graph. The node and edge statistics describe the processed inputs in this study. We retain the released temporal graph samples and their node budgets: 20 for MIT, 50 for Infectious, 60 for Dblp, Highschool, and Tumblr, and 100 for Facebook. Each sample contains an induced temporal contact sequence and a susceptible-infected (SI) dissemination process. The binary classification task distinguishes dissemination processes with infection probabilities $p_1 = 0.2$ and $p_2 = 0.8$. This setup effectively evaluates the model's ability to capture temporal and structural patterns driving dissemination dynamics. Node features are the time-varying infection indicators $x_v(t) = \mathbf{1}\{v \text{ is infected at } t\} \in \{0, 1\}$. Each process starts with one infected node and ends when the infected fraction reaches 0.5; the two infection-probability classes are approximately balanced. The model therefore observes both temporal contacts and evolving infection states. Detailed dataset specifications follow:

- **Dblp** (Oettershagen et al., 2020): Derived from the Dblp bibliography database, this dataset models academic collaboration networks where nodes represent researchers and edges denote joint publications in top-tier machine learning conferences. It consists of 755 dynamic graphs with an average of 53 vertices per graph. The timestamped collaborations provide the contact structure for simulated information dissemination across the academic network.
- **Facebook** (Viswanath et al., 2009): This dataset captures social interactions within the New Orleans Facebook community over a three-month period. Edges represent wall posts or other user interactions. It comprises 995 dynamic graphs with an average of 96 vertices per graph. The interaction times determine the available propagation paths for the simulated dissemination processes across the sampled social network.
- **Highschool** (Oettershagen et al., 2020): Based on data derived from the SocioPatterns project, this network tracks close face-to-face interactions between students in twenty-second intervals over seven consecutive days. The dataset contains 180 dynamic graphs, averaging 52 vertices each. It represents a closed social group for simulating behavioral patterns or information spread among students.
- **Infectious** (Isella et al., 2011): Also sourced from the SocioPatterns project, this graph records physical proximity contacts between visitors at the "Infectious: Stay Away" exhibition. It consists of 200 dynamic graphs with an average of 50 vertices. The contact sequence supports the controlled comparison of dissemination processes with different infection probabilities on the same contact sequence.
- **Tumblr** (Oettershagen et al., 2020): This temporal graph-classification collection tracks information diffusion between Tumblr users. It includes 373 temporal graph samples averaging 53 vertices. These temporal samples support classification of information-diffusion processes in an online community through their observed interaction order.
- **MIT** (Eagle & Pentland, 2006): A subset of the MIT Reality Mining dataset, this temporal graph tracks physical interactions (proximity, location) among 94 MIT faculty and students via 5-minute interval Bluetooth scans and cell tower logs, reflecting real-world social dynamics. The processed benchmark contains 97 induced dynamic subgraphs with an average of 20 vertices. The original study accumulated approximately 450,000 participant-hours of behavioral records over 9 months. This dataset models social network evolution and dissemination process classification in academic communities.

F.3 Baseline details

F.3.1 Baselines for link prediction and node classification

For edge-level and node-level tasks, we select baselines that represent different families of dynamic graph learning approaches, ranging from RNN-based methods to attention-based and ODE-based models commonly adopted in recent temporal-graph benchmarks:

- **TGN** (Rossi et al., 2020): Temporal Graph Networks (TGN) provide a generic framework that combines learnable memory modules with graph neural networks. The framework maintains a dynamic memory state for each node, updated by incoming events, to capture long-term temporal dependencies in continuous time across evolving interaction streams.
- **Jodie** (Kumar et al., 2019): A coupled RNN-based model originally designed for bipartite interaction networks. It learns dynamic latent embeddings of users and items by projecting node states forward in time and predicting their future latent embedding trajectories.
- **DyRep** (Trivedi et al., 2019): A model that captures the interleaving of communication processes and association processes. It utilizes a deep temporal point process to model the occurrence of link events at two different time scales in evolving interaction graphs.
- **DySAT** (Sankar et al., 2020): Dynamic Self-Attention Network employs a joint structural-temporal self-attention mechanism. It computes node latent embeddings by stacking structural attention layers (to capture local neighborhoods) and temporal attention layers (to capture historical evolution).
- **DyGNN** (Ma et al., 2020): This method effectively captures sequential information through streaming updates. An LSTM-based module updates nodes involved in each event, while a propagation mechanism aggregates temporal neighborhood signals. Together, these modules support online prediction from evolving interaction histories.
- **PINT** (Souza et al., 2022): Position-aware Injective Temporal graph neural network addresses the limitations of standard message-passing neural networks (MPNNs) in dynamic settings. It enhances expressiveness by introducing injective temporal message passing and relative positional encoding to capture structural evolution effectively under evolving temporal neighborhoods and irregular event histories.
- **DyGMamba** (Ding et al., 2025): This method adapts state-space models (SSMs) to continuous-time dynamic graphs. It employs node-level and time-level Mamba blocks to capture complex long-range temporal dependencies efficiently while preserving linear computational complexity over long interaction sequences.
- **RAG4DyG** (Wu et al., 2025): A novel framework that reformulates dynamic graph modeling as a retrieval-augmented generation task. It effectively enhances inductive predictions by retrieving and integrating contextually and temporally relevant historical interaction sequences from broader graph structures and historical contexts.
- **DyCIL** (Zhang et al., 2026): This snapshot-based model learns invariant latent embeddings with a dynamic causal subgraph generator, causal-aware attention, and an adaptive environment generator. It supports node classification and link prediction; its environment generator must be trained without access to test labels or future graph structure.
- **LAMP** (De Castelli et al., 2026): This snapshot-based model treats successive graphs as impulses in a neural dynamical system. Antisymmetric propagation and learned dissipation control information retention across nodes and time. Its node latent embeddings support node classification and link prediction with its native temporal-memory mechanism.

F.3.2 Unified assessment protocol for event-driven baselines

For node classification, all temporal models are assessed with the same downstream classifier head. Specifically, after a method produces a node latent embedding $\mathbf{z}_i(t)$, we apply the shared readout so that decoder differences do not dominate the comparison:

$$\hat{y}_i(t) = \mathrm{MLP}(\mathbf{z}_i(t)). \tag{50}$$

This protocol isolates latent embedding quality from task-specific readout engineering. Consequently, the gains reported for IDG are better interpreted as improvements in the latent embedding rather than artifacts of a stronger prediction head or a specially tuned decoder. It also keeps the assessment interface lightweight and reproducible across all evaluated node-classification datasets and supplementary variants considered in the appendix and the extended benchmark comparisons.

Jodie in the unipartite setting. Jodie was originally proposed for bipartite user-item interaction networks, whereas several benchmarks used here are treated as unipartite dynamic graphs after preprocessing. To make the comparison meaningful, we adopt an event-role-based reformulation con-

sistent with the TGN protocol. Every interaction is represented as $(i, j, t, e_{ij}(t))$, both endpoints are drawn from the same node set, and the same recurrent update is shared across nodes:

$$\mathbf{m}_i(t) = \left[\mathbf{s}_j(t^-) \parallel \delta_i(t) \parallel \mathbf{e}_{ij}(t)\right], \tag{51}$$

$$\mathbf{s}_i(t) = \mathrm{RNN}\big(\bar{\mathbf{m}}_i(t), \mathbf{s}_i(t^-)\big). \tag{52}$$

Here, $\mathbf{s}_i(t^-)$ denotes the pre-event state of node i, $\delta_i(t)$ is the elapsed time since its previous update, $\mathbf{e}_{ij}(t)$ is the event feature, and $\bar{\mathbf{m}}_i(t)$ is the aggregated incoming message at time t. In practice, each interaction updates both endpoints with shared parameters, so the model still follows the original event-driven training order even though the user-item semantics disappear. This matters for fairness: otherwise, Jodie would be penalized merely because the benchmark no longer exposes two disjoint node types. Under the reformulated protocol, the comparison isolates whether its recurrent temporal encoding remains competitive once the interface mismatch is removed.

Shared readout for DyRep, DyGMamba, and RAG4DyG. The same MLP readout is also imposed on DyRep, DyGMamba, and RAG4DyG when they are used for node classification. Their task-specific output heads are replaced by the shared readout above, which avoids conflating backbone quality with downstream classifier design. This standardization is important because these baselines were originally paired with different decoders, supervision paths, and output parameterizations. Without a unified head, node-classification performance would mix two effects: the quality of the latent embedding itself and the inductive bias of the final predictor. By exporting every method's latent embedding to the same lightweight classifier, we make the comparison representation-centric. Under this unified protocol, the relative gains of IDG more directly indicate that its latent embeddings preserve global topological evolution and nonlinear feature dependence more effectively than methods whose downstream heads vary across benchmarks.

F.3.3 Baselines for graph classification

For graph-level tasks, we benchmark against temporal graph kernels (Oettershagen et al., 2020), which transform dynamic graphs into static representations using specific strategies and then apply graph kernels for graph-level prediction throughout the comparisons reported here.

Transformation strategies.

- **Reduced Representation (RD)**: Retains the earliest occurrence of each edge and encodes temporal information through labels. This compact static graph loses later repeated interactions and some temporal-walk information.
- **Directed line graph expansion (DL)**: Represents each temporal interaction by two oriented vertices. Directed edges connect compatible interactions in strictly increasing time. This preserves temporal walks but can require quadratically many edges in the expanded static representation of long interaction sequences.
- **Static Expansion (SE)**: Uses time/vertex copies, interaction edges, and waiting edges. This directed acyclic construction encodes temporal progression, waiting times, and connections between successive temporal copies, with size linear in the number of interactions.

Kernels. The strategies above are paired with two standard kernels:

- **Random Walk (RW)**: Compares graphs by counting matching random walks.
- **Weisfeiler-Lehman (WL)**: Compares graphs based on the iterative refinement of node labels, also referred to as colors in the kernel literature.

This yields six distinct baseline kernel combinations: **RD-RW, RD-WL, DL-RW, DL-WL, SE-RW,** and **SE-WL**. Furthermore, we incorporate **APPR-S**, a highly efficient approximation of the DL-RW kernel that leverages sampled temporal random walks to reduce computation while still retaining useful temporal transition statistics for graph-level prediction at larger graph scales.

F.4 Task definitions

In this section, we provide the formal definitions for the downstream tasks and correlation analysis protocols used throughout our experiments and supplementary assessments.

Link prediction. Given a dynamic graph $\mathcal{G} = \{G^1, G^2, \ldots, G^T\}$, the link prediction task aims to predict whether an edge exists between any pair of nodes in the future snapshot G^{t+1} (where $t+1 \leq T$), based on the historical structure and node interaction information observed from the first t snapshots ($1 \leq t < T$) in the evolving graph history.

Node classification. Given a dynamic graph $\mathcal{G} = \{G^1, G^2, \ldots, G^T\}$, the node classification task aims to predict the unlabeled node categories in the target snapshot G^{t+1} (for $t+1 \leq T$), leveraging the evolving structural information, node features, and available partial node labels from the first t snapshots ($1 \leq t < T$) in the observed dynamic sequence.

Graph classification. Each sample is a temporal graph trajectory $\mathcal{G} = \{G^1, \ldots, G^T\}$ with one graph-level label. Under the SI-dissemination protocol, the label identifies the infection-probability class that generated the trajectory. The predictor uses a pooled latent embedding of the observed trajectory; the task is not next-snapshot node or link prediction.

Correlation analysis. Correlation analysis is a statistical assessment method used to quantify the strength and direction of the association between two quantitative variables. We employ the Pearson correlation coefficient (r) to measure linear dependence. Formally, for two variables X and Y with n samples, r is defined as follows in our assessment setup for all reported downstream experiments:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Here (x_i, y_i) pairs one model's intrinsic score and AUC, $\bar{x}, \bar{y}$ are their sample means, and n is the number of models. We report $R^2 = r^2$, which equals the coefficient of determination of a univariate least-squares fit with an intercept. A positive association is checked from the sign of r, not inferred from its square. For constant-score columns, Pearson correlation is undefined because the denominator is zero; such columns are excluded from correlation summaries.

G METRIC ANALYSIS AND CONTROLLED EXPERIMENTS

This section examines estimator accuracy, distribution shift, temporal granularity, and computational cost. Each study reports its comparison setting and evaluation interface. The resulting evidence characterizes latent embedding quality under these protocols; the granularity experiment jointly changes aggregation, physical history, and the number of optimization blocks per epoch.

G.1 SNAPSHOT WINDOWS AND TASK INTERFACES

Before dataset-specific removal of empty windows, interval j contains timestamps in $[b_{j-1}, b_j)$, for strictly increasing boundaries $b_0, \ldots, b_J$. Here J counts physical time intervals. The first boundary is the first retained timestamp, and the final boundary lies beyond the last retained timestamp. Each internal boundary belongs to the following interval. The index t enumerates the resulting processed snapshots, while the neighbor horizon counts these snapshot indices. Tgbl-Wiki discards empty intervals; Email-Eu-core temporal and Sx-MathOverflow retain empty daily intervals and the final partial day within their chronological sequences.

Event streams are sorted chronologically and assigned to contiguous windows. For GitHub, Mooc, Wikipedia, and Reddit, the complete time span of retained events is divided into a fixed number of windows: 15 for GitHub and 20 for each of the other datasets. These dataset-level boundaries remain unchanged across training, validation, and test; each encoder receives only its observed prefix. Tgbl-Wiki models share observed prefixes and prediction cutoffs: snapshot encoders consume aggregated windows, while event encoders retain chronological interactions within those prefixes. For Mooc, the data retain directed user/activity roles and IDG uses a symmetric dot-product link decoder with random negatives and binary cross-entropy. Evaluation inputs precede the prediction target. Tgbl-Wiki AUC/AP describe the paper's candidate-based prediction evaluation; the node-affinity results separately use official NDCG@10.

G.2 Complementary ablations and independent models

The ablations remove SMI, FMI, or the information bottleneck separately, and remove SMI and FMI jointly. The joint removal has the largest degradation, while structural and feature perturbations selectively reduce the corresponding metric. We also recomputed the metric/task correlations with independently trained backbone instances and obtained the same qualitative ranking. These checks support the complementary diagnostic roles of SMI and FMI under the evaluated protocols.

The two scores define complementary diagnostic coordinates. High SMI and FMI indicate retained structural dependence together with attribute dependence. High SMI with lower FMI motivates inspecting the feature pathway, whereas lower SMI with high FMI motivates inspecting structural information loss. Low values of both identify little detected input and latent embedding dependence at the chosen assessment resolution. Their relevance depends on the downstream target, since task-irrelevant information can also be preserved.

G.3 Finite-sample and risk conditions

The FMI decomposition separates critic approximation, sampling, and optimization errors. Conditional on a fixed trajectory and encoder, independent draws with replacement from its empirical record distribution support finite-sample assessment of that distribution. Generalization to unseen trajectories additionally depends on the temporal sampling law. The bottleneck factorization follows the encoder's causal update order, and Proposition 3.1 applies to continuous attributes through conditional mutual information.

G.4 Efficiency boundary

The RTX 3090 runtimes in Tables 3, 18, and 20 measure the per-epoch training pipeline, including spectral assessment, encoding, critic fitting, and the prediction head. The spectral calculation uses joint and marginal Gram matrices with a fixed paired-record budget. Equation 12 accounts for these operations; Appendix A.4 separately accounts for preprocessing and optimization passes.

G.5 Application scope

IDG is designed for sparse, attributed graph sequences with persistent node identities and meaningful snapshot windows. Suitable applications include user/content interaction prediction, future preference ranking, and temporal diffusion classification. SMI and FMI supplement task-based evaluation by quantifying detected structural and attribute dependence. The Genre/Token gains, perturbation responses, and cross-model associations support this diagnostic use within the evaluated model cohort; target-domain validation remains necessary for model selection.

The present encoder operates on pairwise single-relation interactions. Its structural descriptor summarizes symmetrized neighborhoods, and the window size determines temporal resolution. Applications requiring relation-specific semantics, higher-order hyperedges, or exact within-window event order need a representation that preserves those objects. Runtime and ranking comparisons also depend on workload: IDG is slower than several baselines on GitHub and trails LAMP and DyCIL on Tgbn-Reddit. These measured trade-offs guide its use alongside the intrinsic diagnostics.

G.6 Intrinsic metric controls

Common representation interface. The comparison uses the same eleven frozen encoders, five link-prediction datasets, record draws, and final-layer latent embeddings as Figure 4. Each diagnostic produces one score per model and dataset. Pearson R^2 and Spearman ρ then quantify association with the corresponding AUC means. Table 6 uses one (R^2, ρ) pair per cell. The diagnostics assess the same latent embeddings through dependence, effective rank, and controls for latent embedding scale and dimension.

Computable controls. For the record-by-coordinate latent embedding matrix Z, RankMe uses the entropy of normalized singular values (Garrido et al., 2023). With singular values σ_j, define $p_j = \sigma_j / \sum_k \sigma_k$ and score $\exp(-\sum_j p_j \log p_j)$, with zero score for the zero matrix. For

Table 6: Intrinsic diagnostic comparison: each cell contains Pearson R^2 and Spearman ρ with AUC across the same eleven models. Each statistic is ranked separately within its dataset.

Diagnostic	Reddit	Mooc	GitHub	Wikipedia	Tgbl-Wiki
SMI	$\underline{0.797}/\mathbf{0.984}$	$\mathbf{0.909}/\underline{0.927}$	$\mathbf{0.874}/\mathbf{0.986}$	$\underline{0.816}/\mathbf{0.961}$	$\underline{0.804}/\underline{0.973}$
FMI	$\mathbf{0.918}/\underline{0.975}$	$\underline{0.815}/\mathbf{0.945}$	$\underline{0.808}/\underline{0.975}$	$\mathbf{0.922}/\underline{0.909}$	$\mathbf{0.813}/\mathbf{1.000}$
RankMe	$0.493/0.779$	$0.587/0.745$	$0.461/0.656$	$0.552/0.673$	$0.518/0.700$
Structural HSIC	$0.692/0.920$	$0.771/0.845$	$0.724/0.907$	$0.638/0.764$	$0.681/0.682$
Feature HSIC	$0.731/0.743$	$0.673/0.845$	$0.657/0.551$	$0.762/0.682$	$0.728/0.882$
Latent embedding width	$0.005/0.320$	$0.036/-0.075$	$0.020/-0.038$	$0.164/-0.238$	$0.111/0.203$
Mean latent embedding norm	$0.093/0.355$	$0.037/0.064$	$0.071/0.128$	$0.056/-0.100$	$0.119/0.309$

unnormalized Gaussian Gram matrices K_U, K_Z on the same b records, the empirical HSIC score is $\mathrm{Tr}(K_U C K_Z C)/(b-1)^2$, where $C = I_b - \mathbf{1}\mathbf{1}^\top/b$ centers the records (Gretton et al., 2005). Structural HSIC uses the SMI descriptors and kernels; feature HSIC uses normalized attribute vectors and bandwidth 1. SMI and both HSIC controls share the normalized latent embedding kernel. Additional controls are the exported width d_{enc} and mean raw latent embedding norm $b^{-1}\sum_i \|z_i\|_2$. All controls are computed from the frozen latent embeddings without additional encoder training. The exported widths, in the model order of Table 1, are $(128, 64, 128, 256, 128, 128, 128, 256, 128, 128, 128)$ and are fixed across datasets.

Comparison with intrinsic controls. Table 6 shows that both SMI and FMI exceed every control in Pearson R^2 and Spearman ρ on each dataset. Their mean R^2 values are 0.840 and 0.855, compared with 0.522 for RankMe, 0.701 for structural HSIC, and 0.710 for feature HSIC. Relative to the strongest control within each dataset, SMI improves R^2 by 0.054 to 0.150 and FMI by 0.044 to 0.187. By Pearson R^2, SMI is strongest on Mooc and GitHub, whereas FMI is strongest on Reddit, Wikipedia, and Tgbl-Wiki. The two information measures therefore provide complementary diagnostics across the evaluated graph sequences.

Latent embedding width and mean latent embedding norm have mean R^2 of 0.067 and 0.075, respectively, with small rank correlations of mixed sign. By comparison, the mean Spearman correlations are 0.966 for SMI and 0.961 for FMI, versus 0.711 for RankMe, 0.824 for structural HSIC, and 0.741 for feature HSIC. These results favor dependence-sensitive assessment over width, magnitude, or effective rank alone for this model cohort. The paired structure and attribute interfaces also identify which input component is associated with a latent embedding's downstream utility.

Robustness of the descriptive association. For each metric and dataset, Spearman correlation uses average ranks for ties. Leaving out each of the eleven models in turn gives an eleven-value sensitivity range for Pearson R^2. Both calculations use the reported model means and retain the dataset as the comparison unit. Positive affine normalization preserves both correlation measures.

From the reported scores, the mean Spearman correlations are 0.966 for SMI and 0.961 for FMI. Across all five datasets and all eleven single-model exclusions, the Pearson R^2 ranges are 0.769 to 0.954 for SMI and 0.703 to 0.968 for FMI. These descriptive checks retain a positive association under rank comparison and removal of individual model points.

G.7 FMI FINITE-SAMPLE AND FUNCTION-CLASS STUDY

We use a correlated Gaussian sanity check for which the analytic mutual information is

$$I_{\mathrm{true}}(X;Z) = -\frac{d}{2}\log(1-\rho^2).$$

Here d is the feature dimension and ρ the correlation within each of d independent Gaussian coordinate pairs. We fix $d = 20$ and $\rho = \sqrt{1-e^{-2/d}}$ so that $I_{\mathrm{true}} = 1.0$, keeping the critic and optimization protocol fixed while varying sample count. Values are means $\pm$ standard deviations over repeated runs; total gap is I_{true} minus test FMI. The reference objective is estimated from large samples and multiple initializations. The reference gap measures its difference from known synthetic MI; the fit gap compares it with the reported FMI. These empirical gaps combine approximation, sampling, and optimization effects. Moderate capacity improves the fitted objective here, whereas greater capacity can increase optimization variability.

Table 7: Finite-sample FMI convergence. Higher test FMI and lower gaps are better.

n_{sample}	Test FMI	Total gap	Train/test gap	Violations
1,024	0.014 ± 0.023	0.986 ± 0.023	0.563 ± 0.354	0
4,096	0.365 ± 0.014	0.635 ± 0.014	0.566 ± 0.069	0
16,384	0.569 ± 0.010	0.431 ± 0.010	0.166 ± 0.012	0
65,536	$\mathbf{0.625 \pm 0.011}$	$\mathbf{0.375 \pm 0.011}$	$\mathbf{0.045 \pm 0.009}$	0

Table 8: Finite function-class effects at $n = 16,384$. Higher reference/FMI scores and lower gaps are better.

Critic	Params	I_{Θ}^{ref}	Ref. gap	FMI	Fit gap
Bilinear	401	0.870 ± 0.004	0.130 ± 0.004	0.816 ± 0.011	0.055 ± 0.012
MLP-small (32)	1,345	0.628 ± 0.034	0.372 ± 0.034	0.277 ± 0.016	0.351 ± 0.033
MLP-medium (128)	5,377	0.896 ± 0.012	0.104 ± 0.012	0.677 ± 0.012	0.219 ± 0.019
MLP-large (512)	284,161	0.832 ± 0.038	0.168 ± 0.038	0.451 ± 0.128	0.381 ± 0.139

Table 9: DIDA metric and AUC changes from ID to shifted domains.

Dataset	d_{struct}	d_{feature}	SMI: ID→Shift	FMI: ID→Shift	AUC: ID→Shift
COLLAB	0.486 ± 0.037	0.294 ± 0.031	$0.782 \rightarrow 0.603$	$0.704 \rightarrow 0.626$	$0.897 \rightarrow 0.798$
Yelp	0.347 ± 0.042	0.468 ± 0.053	$0.731 \rightarrow 0.645$	$0.773 \rightarrow 0.584$	$0.786 \rightarrow 0.719$
Link-Synthetic	0.001 ± 0.001	0.672 ± 0.048	$0.756 \rightarrow 0.752$	$0.836 \rightarrow 0.493$	$0.905 \rightarrow 0.702$

Table 10: Pearson correlations with AUC in ID and shifted domains.

Dataset	SMI-ID	SMI-Shift	FMI-ID	FMI-Shift
COLLAB	0.861 ± 0.026	0.792 ± 0.039	0.752 ± 0.041	0.684 ± 0.055
Yelp	0.779 ± 0.034	0.718 ± 0.047	0.873 ± 0.023	0.806 ± 0.035
Link-Synthetic	0.241 ± 0.083	0.196 ± 0.097	0.914 ± 0.018	0.857 ± 0.029
Pooled	0.704 ± 0.026	0.642 ± 0.037	0.848 ± 0.019	0.782 ± 0.028

G.8 Distribution shift and pooled correlation

We freeze five DySAT encoders and evaluate DIDA (Zhang et al., 2022) on COLLAB, Yelp, and Link-Synthetic under structural/feature shifts. Here d_{struct} is the Jensen-Shannon spectral distance between domains, d_{feature} is their feature RBF-MMD, and $T_{\text{ID}}, T_{\text{shift}}$ count the in-distribution and shifted evaluation windows. COLLAB and Link-Synthetic use $T_{\text{ID}}/T_{\text{shift}} = 5/5$, while Yelp uses $8/8$. For Link-Synthetic, topology and test windows are fixed and only the feature/label pseudo-correlation changes, which isolates a feature-side shift. The reported changes separate relative metric drops from absolute AUC decreases: COLLAB changes by $-22.9\%/-11.1\%/-0.099$ for SMI/FMI/AUC, Yelp by $-11.8\%/-24.5\%/-0.067$, and Link-Synthetic by $-0.5\%/-41.0\%/-0.203$. In the pooled comparison, the SMI/AUC and FMI/AUC correlations decrease by 0.062 and 0.066, respectively; the per-dataset COLLAB decreases are 0.069 and 0.068. The corresponding SMI/FMI decreases are $0.061/0.067$ on Yelp and $0.045/0.057$ on Link-Synthetic. The pooled window counts are $18/18$. The shifted-domain correlations remain positive across the three evaluated settings.

G.9 Temporal granularity

For Tgbl-Wiki, all four settings share raw events, chronological splits, model parameters, the epoch cap and stopping rule, negative sampling, and five random seeds. We use the half-open timestamp intervals defined in Appendix G.1 and exclude empty windows after aggregation. The remaining windows are indexed consecutively as processed snapshots. The number of 32-window optimization blocks per epoch therefore varies with the retained snapshot count; equal epoch limits do not imply equal update counts. Table 11 shows equal mean AUC at 1 and 60 seconds, with AP decreasing by 0.002. At 300 seconds, AUC/AP decrease by 0.004/0.006, and the intrinsic-score ratios remain 0.979/0.980. At 900 seconds, AUC/AP decrease by 0.017/0.019 and both intrinsic scores fall by approximately 7%. The recurrent state carries information across windows, while the 32-window

Table 11: Tgbl-Wiki temporal granularity. Intrinsic scores are ratios to the 1-second setting; AUC/AP are mean ± standard deviation over five runs.

Δ_{win} (s)	Snapshots	SMI/SMI$_{1s}$	FMI/FMI$_{1s}$	AUC	AP
1	152,757	**1.000**	**1.000**	**0.955±0.002**	**0.961±0.002**
60	42,561	0.993±0.009	0.994±0.010	**0.955±0.003**	0.959±0.003
300	8,929	0.979±0.011	0.980±0.011	0.951±0.003	0.955±0.003
900	2,978	0.933±0.015	0.932±0.017	0.938±0.004	0.942±0.004

candidate cache supplies recent neighbors. Fine windows therefore preserve update order while retaining historical context in the state. Coarsening changes both aggregation and the physical duration of the neighbor horizon and gradient block, with the event stream and chronological split held fixed.

G.10 Independent-model correlation and component profiling

Across the eleven-model comparison, average R^2 is $0.840/0.855$ for SMI/FMI. Excluding IDG gives $0.848/0.832$ across ten baselines, so the positive association also holds across the ten baselines. Each value below is recomputed from Tables 1 and 16; rounding of the published input means limits the reported precision. Across individual datasets, SMI retains R^2 from 0.778 to 0.914 and FMI from 0.762 to 0.922. SMI has the higher association on Mooc and GitHub, and FMI on the remaining datasets. This ordering agrees with the full model cohort. Removing IDG therefore preserves both positive association and the complementary dataset preferences of the two diagnostics.

Table 12: Pearson R^2 between intrinsic metrics and AUC after excluding IDG.

Dataset	SMI/AUC R^2	FMI/AUC R^2
Reddit	0.881	0.922
Mooc	0.880	0.762
GitHub	0.914	0.779
Wikipedia	0.789	0.909
Tgbl-Wiki	0.778	0.787
Average	0.848	0.832

Table 13: Single-epoch component profiling on four additional large-scale datasets (seconds).

Dataset	SMI	FMI	GIB
Tgbn-Genre	18.62 ± 0.71	3.84 ± 0.19	0.61 ± 0.04
Tgbn-Token	286.40 ± 5.82	64.70 ± 1.91	7.90 ± 0.31
Email-Eu-core temporal	92.40 ± 2.85	17.80 ± 0.64	2.40 ± 0.12
Sx-MathOverflow	612.80 ± 14.31	102.50 ± 3.26	11.80 ± 0.47

The component ordering SMI > FMI > GIB identifies the dominant measured cost; GIB timing excludes sampling, encoder, and prediction-head costs. On Tgbn-Token, SMI, FMI, and GIB account for approximately 38.3%, 8.6%, and 1.1% of the 748.35-second full IDG epoch. This ordering identifies spectral assessment as the principal component for reducing the measured computation cost per epoch of IDG under the shared hardware configuration.

G.11 Extended perturbation evaluation

Paired evaluation protocol. We evaluate Reddit, Mooc, GitHub, Wikipedia, and Tgbl-Wiki under structural rewiring and attribute permutation at strengths 5%, 10%, and 20%. Each condition pairs clean and perturbed evaluation of five frozen checkpoints, with three perturbation seeds per checkpoint. Structural rewiring preserves edge count, endpoint types, and node degrees. Attribute permutation exchanges the selected attribute records within the same observation window, using the record interface in Appendix A.7. The encoder parameters, task targets, evaluation candidates, preprocessing, and metric settings are shared by each clean/perturbed pair.

For a metric s, each paired relative change is $100(s_{\text{perturbed}} - s_{\text{clean}})/s_{\text{clean}}$; AUC and AP use the absolute difference $s_{\text{perturbed}} - s_{\text{clean}}$. Table 14 reports the mean and sample standard deviation of the resulting 15 paired differences. These summaries describe joint checkpoint and perturbation variability, with the checkpoint as the model replication unit. The degree-preserving rewiring matches the earlier 10% structural setting. Here the feature intervention is defined over attribute records; Appendix H.5 reports the separate node-permutation setting.

Table 14: Paired perturbation responses over five checkpoints and three perturbation seeds per checkpoint (mean ± sample standard deviation). SMI/FMI changes are relative percentages; AUC/AP changes are absolute. R denotes degree-preserving structural rewiring and P denotes attribute-record permutation. AUC/AP ranks compare retention across strengths within each dataset and perturbation type; larger changes indicate smaller losses.

Dataset	Type	Strength	ΔSMI (%)	ΔFMI (%)	ΔAUC	ΔAP
Reddit	R	5%	-11.80 ± 0.85	-1.80 ± 0.21	$\mathbf{-0.0082 \pm 0.0013}$	$\mathbf{-0.0094 \pm 0.0014}$
		10%	-24.70 ± 1.65	-4.10 ± 0.39	-0.0181 ± 0.0024	-0.0203 ± 0.0026
		20%	-42.60 ± 2.76	-7.80 ± 0.71	-0.0365 ± 0.0043	-0.0412 ± 0.0050
	P	5%	-0.50 ± 0.09	-6.00 ± 0.81	$\mathbf{-0.0047 \pm 0.0008}$	$\mathbf{-0.0058 \pm 0.0010}$
		10%	-1.00 ± 0.17	-12.40 ± 1.48	-0.0102 ± 0.0015	-0.0127 ± 0.0018
		20%	-2.10 ± 0.31	-24.80 ± 2.54	-0.0221 ± 0.0027	-0.0276 ± 0.0034
Mooc	R	5%	-13.70 ± 1.06	-1.60 ± 0.21	$\mathbf{-0.0108 \pm 0.0016}$	$\mathbf{-0.0091 \pm 0.0014}$
		10%	-29.40 ± 1.96	-3.80 ± 0.36	-0.0242 ± 0.0029	-0.0208 ± 0.0027
		20%	-47.90 ± 3.33	-7.30 ± 0.66	-0.0475 ± 0.0052	-0.0421 ± 0.0046
	P	5%	-0.60 ± 0.10	-4.10 ± 0.62	$\mathbf{-0.0014 \pm 0.0004}$	$\mathbf{-0.0011 \pm 0.0003}$
		10%	-1.20 ± 0.21	-9.00 ± 1.24	-0.0030 ± 0.0007	-0.0025 ± 0.0006
		20%	-2.50 ± 0.36	-17.90 ± 2.07	-0.0065 ± 0.0012	-0.0055 ± 0.0011
GitHub	R	5%	-15.10 ± 1.08	-2.10 ± 0.24	$\mathbf{-0.0113 \pm 0.0017}$	$\mathbf{-0.0178 \pm 0.0020}$
		10%	-31.80 ± 2.05	-4.80 ± 0.47	-0.0260 ± 0.0030	-0.0420 ± 0.0037
		20%	-49.70 ± 3.87	-8.50 ± 0.80	-0.0492 ± 0.0057	-0.0753 ± 0.0067
	P	5%	-0.35 ± 0.08	-4.60 ± 0.59	$\mathbf{-0.0005 \pm 0.0002}$	$\mathbf{-0.0025 \pm 0.0006}$
		10%	-0.80 ± 0.14	-8.70 ± 1.05	-0.0010 ± 0.0004	-0.0053 ± 0.0011
		20%	-1.70 ± 0.24	-17.10 ± 1.98	-0.0020 ± 0.0006	-0.0115 ± 0.0020
Wikipedia	R	5%	-10.30 ± 0.83	-1.70 ± 0.19	$\mathbf{-0.0030 \pm 0.0011}$	$\mathbf{-0.0055 \pm 0.0012}$
		10%	-21.90 ± 1.46	-3.70 ± 0.35	-0.0070 ± 0.0018	-0.0130 ± 0.0021
		20%	-38.40 ± 2.64	-7.20 ± 0.59	-0.0151 ± 0.0034	-0.0268 ± 0.0041
	P	5%	-0.40 ± 0.09	-6.10 ± 0.95	$\mathbf{-0.0028 \pm 0.0007}$	$\mathbf{-0.0048 \pm 0.0010}$
		10%	-0.80 ± 0.15	-12.90 ± 1.73	-0.0063 ± 0.0012	-0.0103 ± 0.0019
		20%	-1.80 ± 0.27	-25.70 ± 2.88	-0.0138 ± 0.0022	-0.0227 ± 0.0031
Tgbl-Wiki	R	5%	-9.70 ± 0.75	-1.40 ± 0.16	$\mathbf{-0.0035 \pm 0.0007}$	$\mathbf{-0.0028 \pm 0.0007}$
		10%	-19.80 ± 1.45	-3.00 ± 0.28	-0.0080 ± 0.0014	-0.0067 ± 0.0013
		20%	-35.60 ± 2.57	-5.90 ± 0.53	-0.0180 ± 0.0025	-0.0153 ± 0.0024
	P	5%	-0.50 ± 0.10	-7.10 ± 0.96	$\mathbf{-0.0065 \pm 0.0011}$	$\mathbf{-0.0059 \pm 0.0009}$
		10%	-1.10 ± 0.18	-14.50 ± 1.79	-0.0144 ± 0.0019	-0.0129 ± 0.0017
		20%	-2.30 ± 0.33	-28.80 ± 3.16	-0.0318 ± 0.0035	-0.0282 ± 0.0031

Selective metric response. Structural rewiring produces a larger SMI decrease than FMI decrease on all five datasets at every strength. At 20%, SMI decreases by 35.60% to 49.70%, compared with 5.90% to 8.50% for FMI. Attribute permutation gives the complementary pattern: FMI decreases by 17.10% to 28.80%, while SMI changes by only 1.70% to 2.50%. Both responses grow monotonically with perturbation strength, showing that the paired diagnostics distinguish topology changes from attribute changes under these controlled interventions.

Downstream response and application relevance. Every tested perturbation lowers mean AUC and AP, with larger losses at higher strengths. At 20% rewiring, GitHub has the largest AUC/AP decreases, 0.0492/0.0753; its attribute-permutation decreases are 0.0020/0.0115. Mooc similarly responds more strongly to rewiring than to attribute permutation. Tgbl-Wiki shows the opposite downstream ordering at 20%: attribute permutation lowers AUC/AP by 0.0318/0.0282, versus 0.0180/0.0153 for rewiring. These dataset-specific responses support using SMI and FMI together to diagnose topology-sensitive and attribute-sensitive prediction workloads.

G.12 TASK-INFORMATION DECOMPOSITION AND READOUT RISK

Proposition 3.1 applies to finite task labels with arbitrary measurable input and latent spaces. Fix the learned encoder before drawing evaluation records. Its stochastic noise is independent of the target conditional on the observed input, giving $Y \to D \to Z$. This is the encoder's causal sampling rule and places no independence condition on different coordinates of the raw graph.

Information and readout errors. For any normalized readout q, expanding the conditional KL gives the exact identity

$$\begin{aligned}\mathcal{R}_{\log}(q \circ Z) &= H(Y \mid Z) + \mathbb{E}_Z D_{\mathrm{KL}}(P(Y \mid Z) \| q(Y \mid Z)) \\ &= H(Y \mid D) + I(Y; D \mid Z) + \mathbb{E}_Z D_{\mathrm{KL}}(P(Y \mid Z) \| q(Y \mid Z)).\end{aligned} \tag{53}$$

All three terms are nonnegative for a finite-valued target. The second term is zero exactly when the representation is sufficient for predicting Y from D up to null sets. Its chain-rule decomposition in equation 11 allows structural and feature information to interact: the feature term is conditional on both the observed structure and representation. Thus the analysis accommodates continuous node attributes and coupled structural and feature effects.

Continuous neural prediction classes. Let $\mathcal{Q}$ be a fixed class of neural readouts with C_Y target classes. Probability smoothing defines $q_\epsilon(y \mid z) = (1 - C_Y\epsilon)q(y \mid z) + \epsilon$ for $0 < \epsilon < 1/C_Y$, so the normalized probabilities are at least ϵ and log-loss lies in $[0, B]$, with $B = \log(1/\epsilon)$. Define the loss class $\mathcal{F}_\mathcal{Q} = \{(z, y) \mapsto -\log q_\epsilon(y \mid z) : q \in \mathcal{Q}\}$. For m independent readout records $S = ((z_i, y_i))_{i=1}^m$, its expected Rademacher complexity is

$$\mathrm{Rad}_m(\mathcal{F}_\mathcal{Q}) = \mathbb{E}_{S,\varepsilon} \sup_{f \in \mathcal{F}_\mathcal{Q}} \frac{1}{m} \sum_{i=1}^m \varepsilon_i f(z_i, y_i),$$

where the independent signs ε_i are uniformly -1 or $+1$. This quantity is defined for infinite neural classes and quantifies their capacity on the evaluation distribution (Mohri et al., 2018).

Symmetrization and bounded differences give, with probability at least $1 - \delta$,

$$\sup_{q \in \mathcal{Q}} |\widehat{\mathcal{R}}(q_\epsilon) - \mathcal{R}(q_\epsilon)| \le 2\,\mathrm{Rad}_m(\mathcal{F}_\mathcal{Q}) + B\sqrt{\frac{\log(2/\delta)}{2m}}.$$

Let $\widehat{q}$ minimize empirical risk to tolerance $\eta_{\mathrm{opt}} \ge 0$ and set $\alpha_\mathcal{Q} = \inf_{q \in \mathcal{Q}} \mathcal{R}(q_\epsilon \circ Z) - H(Y \mid Z)$. Insert the empirical risk of $\widehat{q}$ and that of the best population readout; applying the uniform deviation twice yields

$$\begin{aligned}\mathcal{R}(\widehat{q}_\epsilon \circ Z) \le\ & H(Y \mid D) + I(Y; D \mid Z) + \alpha_\mathcal{Q} + \eta_{\mathrm{opt}} \\ & + 4\,\mathrm{Rad}_m(\mathcal{F}_\mathcal{Q}) + 2B\sqrt{\frac{\log(2/\delta)}{2m}}.\end{aligned} \tag{54}$$

The approximation term includes the smoothing and readout-class restrictions. The optimization tolerance explicitly accommodates approximate neural training. For a fixed finite collection of independent readout records this is a standard capacity bound, independent of the dimensionality or differential entropy of the attributes.

Practical interpretation. The identity in equation 53 holds at the population level without an independence assumption between graph snapshots. The finite-sample bound concerns independent records sampled after encoder fitting, such as independent graph trajectories. A dependent event stream requires a corresponding block-dependent uniform-deviation bound. This distinction separates an exact information identity from the sampling requirements of a generalization statement.

SMI and FMI supply label-free structural and attribute diagnostics, whereas $I(Y; D \mid Z)$ measures task-specific information loss. Strong input dependence can include nuisance information, and a compact task-sufficient representation can intentionally discard it. The empirical correlation, perturbation, and sensitivity studies characterize where the diagnostics are useful. The supervised bottleneck balances retained input dependence against predictive information and compression under the same supervised task distribution.

H SUPPLEMENTARY EXPERIMENTS

H.1 GRAPH CLASSIFICATION

Table 15: Graph-classification ACC (mean ± standard deviation). Kernel baselines reproduce the published Table 4 results (Micheli & Tortorella, 2022), converted from percentages; APPR-S uses $S = 250$. **Bold**/underline: best/second-best mean. Δ is the relative gain in six-dataset mean ACC.

ACC Performance							
Baseline	**Dblp**	**Facebook**	**Highschool**	**Infectious**	**Tumblr**	**MIT**	$\Delta(\%)$
RD-RW	0.642±0.012	0.664±0.006	0.801±0.019	0.740±0.015	0.695±0.010	0.606±0.047	↑24.5
RD-WL	0.673±0.005	0.667±0.006	0.839±0.007	0.774±0.012	0.697±0.005	0.692±0.018	↑18.9
DL-RW	0.818±0.009	0.800±0.005	0.934±0.010	**0.887±0.012**	0.772±0.010	0.826±0.021	↑2.5
DL-WL	0.765±0.010	0.800±0.005	0.893±0.007	0.787±0.015	0.782±0.013	0.364±0.040	↑17.6
SE-RW	0.831±0.006	0.802±0.006	0.933±0.013	0.872±0.011	0.782±0.009	0.573±0.033	↑7.7
SE-WL	0.820±0.006	0.753±0.003	0.910±0.012	0.798±0.015	**0.806±0.007**	0.508±0.042	↑12.4
APPR-S	0.764±0.009	0.779±0.006	0.904±0.018	0.786±0.021	0.784±0.013	0.669±0.025	↑10.2
IDG (Ours)	**0.910±0.035**	**0.859±0.035**	**0.943±0.055**	0.883±0.067	0.731±0.083	**0.837±0.113**	-

IDG has the highest reported mean on four of six graph-classification datasets (Table 15). Its gains over the strongest listed baseline are 0.079 ACC on Dblp, 0.057 on Facebook, 0.009 on Highschool, and 0.011 on MIT, or 9.5%, 7.1%, 1.0%, and 1.3%. The table compares IDG with published kernel results under the same ct2 task definition; the resampling protocols are specified in Appendix F.1.

Across all six datasets, IDG obtains mean ACC 0.861, compared with 0.840 for DL-RW, the strongest baseline by this aggregate. IDG reaches 0.883 ACC on Infectious and 0.731 on Tumblr; the corresponding leading kernel scores are 0.887 and 0.806. Together with the four leading IDG results, these values characterize performance across distinct temporal graph families.

H.2 CORRELATION ANALYSIS OF ASSESSMENT METRICS

We compute both intrinsic scores for all eleven models. Dataset-wise min-max normalization maps the lowest and highest score to zero and one. IDG's score of one denotes the cohort maximum. The correlation analysis pairs these normalized scores with link-prediction AUC.

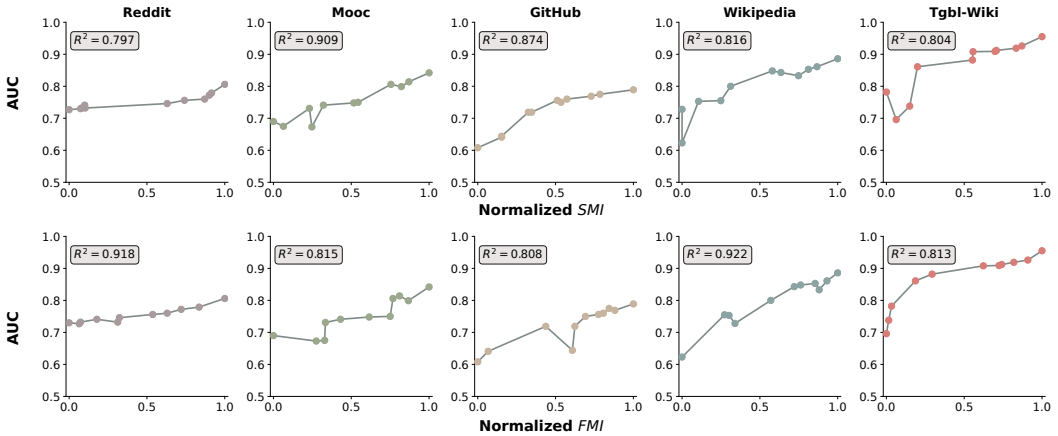

Figure 4: Normalized SMI/FMI versus link-prediction AUC. R^2 quantifies dataset-wise empirical association.

Table 16 lists the normalized scores, and Figure 4 pairs them with AUC from Table 1. The lines connect models in increasing metric order. Both metrics have positive correlations with AUC, with average R^2 of 0.840 for SMI and 0.855 for FMI. SMI has the stronger association on Mooc and GitHub; FMI has the stronger association on Reddit, Wikipedia, and Tgbl-Wiki. This variation

Table 16: Normalized SMI and FMI scores for different models across five datasets in the link prediction task.

Model	Reddit		Mooc		GitHub		Wikipedia		Tgbl-Wiki	
	SMI	FMI	SMI	FMI	SMI	FMI	SMI	FMI	SMI	FMI
Jodie	0.103	0.073	0.231	0.334	0.535	0.692	0.249	0.273	0.150	0.015
DyRep	0.000	0.065	0.000	0.000	0.325	0.437	0.312	0.570	0.063	0.000
DySAT	0.073	0.000	0.247	0.275	0.000	0.000	0.000	0.340	0.553	0.294
DyGNN	0.100	0.178	0.064	0.329	0.153	0.067	0.000	0.000	0.200	0.187
TGN	0.630	0.323	0.321	0.431	0.346	0.624	0.580	0.762	0.000	0.033
PINT	0.080	0.312	0.755	0.768	0.153	0.608	0.106	0.302	0.557	0.623
DyGMamba	0.741	0.537	0.544	0.750	0.511	0.776	0.635	0.721	0.707	0.741
RAG4DyG	0.871	0.630	0.516	0.615	0.573	0.807	0.748	0.881	0.698	0.724
DyCIL	0.900	0.720	0.870	0.810	0.785	0.844	0.812	0.855	0.834	0.819
LAMP	0.915	0.835	0.822	0.867	0.729	0.881	0.867	0.931	0.871	0.909
IDG	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**

supports considering both coordinates during diagnosis. The task-information decomposition separately characterizes the information relevant to prediction.

H.3 Detailed ablation study

IDG contains several complementary modules for preserving dynamic information during long-term graph evolution. To test their contribution, we compare four variants on link prediction. **IDG-No-SMI** sets the SMI weight to zero; **IDG-No-FMI** sets the FMI weight to zero; **IDG-No-IC** sets $\beta = 0$, removing both compression penalties while retaining supervised prediction; and **IDG-No-SMI&FMI** sets $\lambda = 0$. All variants retain the encoder, decoder, prediction loss, and data protocol. These interventions isolate the contributions of the objective terms within the shared architecture.

Figure 5 reports means and error bars for all five datasets. Joint removal of SMI and FMI lowers AUC by 0.046 to 0.169 and AP by 0.052 to 0.228. Removing SMI alone has its largest effect on GitHub (AUC/AP decreases of 0.122/0.193), while removing FMI has its largest effect on Wikipedia (0.134/0.147). The different responses indicate that the two objectives contribute differently across the evaluated datasets.

Removing the bottleneck objective lowers AUC by 0.014 to 0.027 and AP by 0.020 to 0.043. The full model has the highest mean in every panel. On Mooc, the full model reaches 0.803 AP compared with 0.801 after removing FMI, and removing both metrics gives 0.731 AP. The objective ablations quantify complementary structural and feature contributions across datasets.

H.4 Scalability and efficiency

This section evaluates official node-affinity ranking on three TGB datasets and graph-classification accuracy and cost on two additional temporal networks. The node-affinity ranking results use the benchmark task definitions and NDCG@10; the graph experiments use the dissemination-process protocol defined in Appendix F.2.

Table 17: Raw TGB node-affinity datasets. Times count distinct timestamps; prefix Tgbn- is omitted.

Dataset	Nodes	Events	Times
Reddit	11,766	27,174,118	21,889,537
Genre	1,505	17,858,395	133,758
Token	61,756	72,936,998	2,036,524

Official node-affinity ranking task.
Tgbn-Reddit records user/subreddit posts, Tgbn-Genre records weighted user/music genre interactions, and Tgbn-Token records user/token transactions (Huang et al., 2023; Nadiri & Takes, 2022; Bertin-Mahieux et al., 2011; Shamsi et al., 2022). Each task predicts a user's destination-affinity ranking for the next week. Genre weights reflect song/genre membership; Token uses the release's log-normalized transaction weights. Table 17 describes the raw event streams, while computational complexity depends on the causal windows and candidate edges actually processed.



Figure 5: Cross-dataset link-prediction ablations. Labels show means; black error bars show standard deviations. Red bars and bottom labels give the absolute mean gap to IDG.

Chronological evaluation. We use the official dataset splits, node-label release times, target vectors, and destination ordering. At each evaluation time, the encoder consumes only the available interaction prefix. A snapshot window is closed at a prediction boundary so that no later event enters its latent embedding. States carry across successive windows, with truncated backpropagation as specified in Appendix A.4. Validation selects the checkpoint; test labels are used only for scoring. The TGB label-time iterator is reset at the start of each pass. This procedure preserves the temporal boundary between observed interactions and future affinities.

Ranking head and metric. A two-layer node readout produces one score for each destination in the official target vector. We train with cross-entropy against the normalized affinity target, using the same destination set for all models. Evaluation passes the full score and relevance vectors to the official TGB evaluator, which computes NDCG@10 with its prescribed gain and tie handling. There is no binary-label conversion or sampled-negative classification step in this task. The log-loss analysis applies to a categorical destination drawn from the affinity distribution; the reported ranking metric is NDCG@10.

Ranking comparison. IDG reaches 0.406 on Genre and 0.269 on Token, improving over LAMP by 3.6% and 4.3%, respectively. Against DyCIL, the improvements are 0.023 and 0.022 in absolute NDCG@10. Relative to the strongest event-oriented competitor DyGMamba, the gains are 0.028 and 0.031. These results support the use of joint structural and feature assessment for the evaluated affinity tasks. On Reddit, IDG ranks third at 0.338 NDCG@10, following LAMP (0.348) and DyCIL (0.340), and exceeds DyGMamba by 0.006. The results motivate selecting IDG for its measured accuracy and diagnostic value in the target domain.

Table 18: Official TGB node-affinity ranking (NDCG@10, higher is better) and per-epoch time on one RTX 3090 (seconds, lower is better; mean ± standard deviation). **Bold**/underline: best/second-best value.

Method	Tgbn-Reddit		Tgbn-Genre		Tgbn-Token	
	NDCG@10	Time (s)	NDCG@10	Time (s)	NDCG@10	Time (s)
Jodie	0.308	163.20±5.25	0.348	80.42±3.18	0.155	862.17±12.84
DyRep	0.312	190.26±6.21	0.351	96.83±3.76	0.141	931.65±16.77
DySAT	0.322	**131.40±10.13**	0.359	**61.57±5.94**	0.184	781.43±24.66
DyGNN	0.325	1725.91±59.59	0.361	1086.24±37.55	0.197	5128.34±176.52
TGN	0.315	245.39±7.88	0.367	129.73±4.89	0.169	1098.74±20.93
PINT	0.328	1185.31±39.35	0.374	742.18±24.63	0.212	3564.82±118.45
DyGMamba	0.332	343.40±12.80	0.378	221.46±8.35	0.238	963.51±34.92
RAG4DyG	0.319	289.74±10.68	0.356	186.92±7.04	0.191	812.26±28.37
DyCIL	0.340	345.44±13.61	0.383	175.00±7.00	0.247	1710.00±65.00
LAMP	**0.348**	184.04±7.48	0.392	92.00±4.00	0.258	940.00±32.00
IDG	0.338	138.85±2.16	**0.406**	67.84±1.26	**0.269**	**748.35±6.42**

IDG is second-fastest on Reddit and Genre and fastest on Token. On Reddit, its 138.85-second epoch costs 5.7% more than DySAT while improving NDCG@10 by 5.0%. It reduces time by 24.6% relative to LAMP, with a 0.010 lower ranking score. Relative to DySAT, Genre NDCG@10 improves by 13.1% with 10.2% more epoch time; Token NDCG@10 improves by 46.2% while epoch time falls by 4.2%. Compared with LAMP, IDG reduces Genre/Token epoch time by 26.3%/20.4% while improving both scores. These comparisons characterize the measured accuracy and cost trade-offs.

Email-Eu-core temporal dataset (Paranjape et al., 2017): The SNAP temporal e-mail network contains 986 users, 332,334 events, and 24,929 distinct directed sender/recipient pairs. Events are stored as source, destination, and Unix timestamp. The source reports a time span of 803 days. Starting at the first event, one-day windows retain intermediate empty days and the final partial day, giving 804 windows, of which 526 are nonempty. Temporal edge direction is preserved during SI propagation.

Table 19: Large-scale graph datasets.

Metric	Email-Eu-core temporal	Sx-MathOverflow
# Nodes	986	24,818
Events	332,334	506,550
Directed pairs	24,929	239,978
Daily windows	804	2,351
Nonempty windows	526	2,350
Base subgraphs	886	20,785

Sx-MathOverflow dataset (Paranjape et al., 2017): This SNAP network combines answer- and comment-based interactions among 24,818 users. Its 506,550 events contain 239,978 distinct directed pairs over a reported span of 2,350 days. Applying the same daily window convention gives 2,351 windows, including 2,350 nonempty windows. Event counts describe repeated temporal contacts, whereas pair counts describe distinct directed relationships in the aggregate network.

Base-subgraph construction and sample counts. For each network, we start one BFS from every node in the undirected aggregate contact graph. The node budget is $B_{\mathrm{BFS}} = 100$; exploration stops when the budget is reached or the queue is empty. We retain node sets of size at least 20, remove duplicate node sets, and extract all temporal contacts induced by each retained set. There is no additional hop limit; retained subgraphs have BFS depths of 1 to 4. SI propagation then uses the original directed contacts and the infection-state features defined in Appendix F.2.

Email supplies 986 eligible candidates; removing 100 duplicate node sets leaves 886 base subgraphs. MathOverflow supplies 24,818 candidates, of which 24,668 meet the size threshold; removing 3,883 duplicates leaves 20,785 base subgraphs. Table 19 distinguishes sample counts from temporal windows. Every comparator receives the same samples, features, and SI classes. This setting evaluates collections of local graphs with at most 100 nodes each; the full-network statistics describe their source networks.

On the additional graph-classification benchmarks, IDG improves ACC over SE-RW from 0.873 to 0.917 on Email and from 0.851 to 0.892 on MathOverflow. Its per-epoch time is 70.7% and 74.5% lower, respectively. RD-WL and APPR-S remain faster: compared with APPR-S, IDG gains 0.091/0.094 ACC at 3.24/2.87 times the epoch cost. Thus the measured

Table 20: Graph-classification ACC and per-epoch time in seconds on two additional networks (mean ± standard deviation).

Method	Email-Eu-core temporal		Sx-MathOverflow	
	ACC	Time (s)	ACC	Time (s)
RD-RW	0.688±0.009	238.41±9.35	0.661±0.010	1456.33±52.71
RD-WL	0.711±0.008	**84.26±3.18**	0.684±0.009	**512.68±18.64**
DL-RW	0.864±0.006	1548.72±53.41	0.841±0.007	9516.47±318.52
DL-WL	0.838±0.007	612.57±21.84	0.809±0.008	3828.15±136.44
SE-RW	0.873±0.006	1317.64±46.92	0.851±0.007	8174.92±288.73
SE-WL	0.846±0.007	541.89±19.37	0.821±0.008	3392.64±121.36
APPR-S	0.826±0.007	119.43±4.26	0.798±0.008	726.55±25.08
IDG	**0.917±0.005**	386.72±7.14	**0.892±0.006**	2084.37±38.51

advantage is higher accuracy at moderate cost relative to expensive temporal expansions, with a clear speed trade-off against the fastest kernels.

H.5 Post hoc perturbation diagnostics

To further evaluate the effectiveness of SMI and FMI as interpretable diagnostic metrics rather than mere regularization terms, we conduct a post hoc perturbation analysis. The goal is to verify whether these intrinsic metrics can accurately diagnose the source of latent embedding degradation when the input dynamic graph is subjected to specific types of noise.

Table 21: GitHub perturbation diagnostics.

Condition	SMI	FMI	AUC	AP
Clean	**1.000**	**1.000**	**0.789±0.005**	**0.797±0.004**
Struct. pert.	0.682	0.952	0.763±0.006	0.755±0.006
Feature pert.	0.972	0.746	0.786±0.005	0.779±0.005

Table 22: Wikipedia perturbation diagnostics.

Condition	SMI	FMI	AUC	AP
Clean	**1.000**	**1.000**	**0.886±0.004**	**0.913±0.003**
Struct. pert.	0.781	0.963	0.879±0.005	0.900±0.004
Feature pert.	0.972	0.628	0.865±0.007	0.877±0.006

We select the GitHub and Wikipedia datasets under the link prediction setting. After IDG is fully trained, we freeze the model parameters to prevent any optimization adaptation. We then introduce specific perturbations to the test data and record the changes in both the intrinsic metrics (SMI, FMI) and downstream performance (AUC, AP). The evaluation is conducted under three distinct conditions:

1. **Clean:** The original test set without any perturbation.
2. **Structural Perturbation:** We perform degree-preserving rewiring on 10% of the edges in the historical snapshots, while keeping the node features entirely unchanged.
3. **Feature Perturbation:** We randomly shuffle the input features for 30% of the nodes, while leaving the original adjacency matrix and all edge directions unchanged.

The post hoc diagnostic results for GitHub and Wikipedia are presented in Table 21 and Table 22, respectively (SMI and FMI are normalized relative to the clean condition).

On GitHub, structural rewiring reduces SMI by 31.8% and FMI by 4.8%, accompanied by AUC/AP decreases of 0.026/0.042. Feature shuffling reduces FMI by 25.4% and SMI by 2.8%, with smaller AUC/AP decreases of 0.003/0.018. On Wikipedia, feature shuffling produces the larger AUC/AP decreases (0.021/0.036), while rewiring reduces them by 0.007/0.013. Its FMI decreases by 37.2% under feature shuffling, compared with a 21.9% SMI decrease under rewiring. The selective responses identify different failure modes across the two datasets under the shared evaluation protocol.

H.6 Cross-task and cross-dataset sensitivity analysis

In this section, we conduct a sensitivity analysis on the SMI/FMI weight λ (equation 10), the information bottleneck weight β (equation 8), and the neighbor budget k_{nbr} in the local dependence assumption. Figure 6 summarizes Reddit link prediction, Reddit node classification, and Dblp graph classification. The tables extend these sweeps to Wikipedia, GitHub, Facebook, and Highschool, using AUC for link/node prediction and ACC for graph classification.

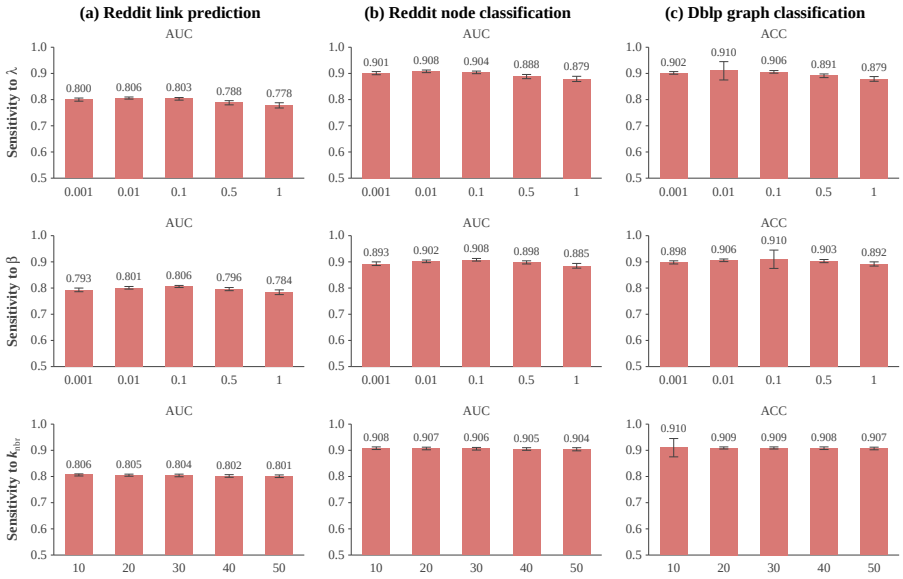

Figure 6: Parameter sensitivity on (a) Reddit link prediction (AUC), (b) Reddit node classification (AUC), and (c) Dblp graph classification (ACC). Rows vary λ, β, and k_{nbr}; bars and error bars show the means and standard deviations for these task and dataset combinations.

Figure 6 shows that IDG is most stable with respect to the neighbor budget k_{nbr}. The scores remain nearly flat across all three tasks, with small decreases or ties as k_{nbr} increases from 10 to 50. This indicates that the model can preserve the essential local structure even with a compact neighbor budget, so the latent embedding does not rely on aggressively enlarging the sampled context.

The sensitivity to λ and β reflects their roles in information preservation and compression. The representative curves in Figure 6 favor $\lambda = 0.01$ and $\beta = 0.1$. The expanded tables identify dataset-specific optima in the moderate-weight range, supporting common initial defaults followed by validation-based selection for individual downstream tasks.

Table 23: Cross-dataset sensitivity for λ across tasks.

λ	Link prediction			Node classification			Graph classification		
	Reddit	Wikipedia	GitHub	Reddit	Wikipedia	GitHub	Dblp	Facebook	Highschool
0.001	0.800±0.006	0.884±0.005	0.794±0.006	0.901±0.006	0.928±0.005	0.868±0.006	0.902±0.005	0.851±0.006	0.938±0.004
0.01	**0.806±0.004**	0.886±0.004	0.789±0.005	**0.908±0.005**	**0.936±0.004**	**0.876±0.005**	**0.910±0.035**	**0.859±0.035**	**0.943±0.055**
0.1	0.803±0.005	**0.887±0.005**	**0.797±0.005**	0.904±0.005	0.931±0.005	0.872±0.005	0.906±0.005	0.854±0.005	0.940±0.004
0.5	0.788±0.008	0.876±0.007	0.783±0.008	0.888±0.008	0.916±0.007	0.856±0.008	0.891±0.007	0.842±0.007	0.931±0.006
1	0.778±0.010	0.868±0.009	0.771±0.010	0.879±0.010	0.905±0.009	0.844±0.010	0.879±0.009	0.829±0.009	0.921±0.008

Table 23 favors moderate metric weights. Seven of the nine task and dataset combinations reach their highest mean at $\lambda = 0.01$; Wikipedia and GitHub link prediction peak at $\lambda = 0.1$. Every combination performs better at its preferred moderate weight than at $\lambda = 1$. These results support validation-based selection within the interval from 0.01 to 0.1, with 0.01 serving as the shared default.

Table 24: Cross-dataset sensitivity for β across tasks.

β	Link prediction			Node classification			Graph classification		
	Reddit	Wikipedia	GitHub	Reddit	Wikipedia	GitHub	Dblp	Facebook	Highschool
0.001	0.793±0.007	0.878±0.006	0.788±0.007	0.893±0.007	0.923±0.006	0.864±0.007	0.898±0.006	0.849±0.006	0.936±0.004
0.01	0.801±0.005	**0.886±0.005**	**0.796±0.005**	0.902±0.005	0.931±0.005	0.872±0.005	0.906±0.005	0.855±0.005	0.940±0.004
0.1	**0.806±0.004**	**0.886±0.004**	0.789±0.005	**0.908±0.005**	**0.936±0.004**	**0.876±0.005**	**0.910±0.035**	**0.859±0.035**	**0.943±0.055**
0.5	0.796±0.006	0.882±0.006	0.792±0.006	0.898±0.006	0.927±0.006	0.868±0.006	0.903±0.006	0.852±0.006	0.938±0.005
1	0.784±0.009	0.873±0.008	0.780±0.009	0.885±0.009	0.915±0.008	0.856±0.009	0.892±0.008	0.842±0.008	0.931±0.006

Table 24 favors moderate compression. Eight of the nine combinations attain their highest mean at $\beta = 0.1$, including a tie with 0.01 for Wikipedia link prediction. GitHub link prediction reaches its highest mean at 0.01. The values 0.01 and 0.1 therefore provide a useful validation range, and the shared default 0.1 performs strongly across the node and graph classification settings.

Table 25: Cross-dataset sensitivity for neighbor budget k_{nbr} across tasks.

k_{nbr}	Link prediction			Node classification			Graph classification		
	Reddit	Wikipedia	GitHub	Reddit	Wikipedia	GitHub	Dblp	Facebook	Highschool
10	**0.806±0.004**	0.886±0.004	0.789±0.005	**0.908±0.005**	**0.936±0.004**	**0.876±0.005**	**0.910±0.035**	**0.859±0.035**	**0.943±0.055**
20	0.805±0.004	**0.890±0.004**	**0.801±0.005**	0.907±0.005	0.935±0.004	0.875±0.005	0.909±0.004	0.858±0.005	**0.943±0.003**
30	0.804±0.005	0.889±0.004	0.800±0.005	0.906±0.005	0.935±0.004	0.874±0.005	0.909±0.004	0.858±0.005	0.942±0.004
40	0.802±0.005	0.888±0.005	0.799±0.005	0.905±0.005	0.934±0.005	0.873±0.005	0.908±0.005	0.857±0.005	0.942±0.004
50	0.801±0.005	0.887±0.005	0.798±0.006	0.904±0.006	0.933±0.005	0.872±0.006	0.907±0.005	0.856±0.005	0.941±0.004

Table 25 shows a narrow score range as the neighbor budget varies from 10 to 50. The largest range is 0.012 AUC on GitHub; Wikipedia spans 0.004, and every node and graph classification column spans at most 0.004. Wikipedia and GitHub link prediction peak at $k_{\mathrm{nbr}} = 20$, while the shared default $k_{\mathrm{nbr}} = 10$ retains strong results across the other tasks. The compact default reduces the local aggregation budget, and validation can select a larger neighborhood for the target dataset.

Tables 23 to 25 provide the detailed tabular version and extension of this sensitivity analysis. Figure 6 uses the Reddit link/node columns and the Dblp graph column of these tables. Seven of nine task and dataset combinations favor $\lambda = 0.01$, and eight favor $\beta = 0.1$. Wikipedia and GitHub link prediction favor a neighbor budget of 20; the remaining combinations favor or tie at 10. These comparisons support moderate regularization with validation-based selection for each task.

I NOTATION REFERENCE

Table 26: Notation. Indices follow their stated sampling distributions.

Symbol	Definition
$G, \mathcal{G}; G_t$	Observed dynamic graph sequence; snapshot at processed index t.
b_j, J	Physical time boundary and number of intervals before removal of empty windows.
V_t, E_t, A_t, X_t	Snapshot vertices, edges, adjacency, and attributes.
$G_e; M, H$	Latent embedding; its structural and continuous feature states.
$\bar{V}, n; T, L$	Common vertex basis and size; prefix length and layer count.
$S^i_{a\alpha}$	Assignment of observed node a to latent node α.
$B^i_{a\alpha}, P_e, \mu$	Assignment prior, edge-error probability, and agreement log-odds.
$a_{v,t}, z^l_{v,t}; d^t_v$	Sparse input structural descriptor, paired latent embedding, and observed degree.
$\mathrm{Enc}, d_{\mathrm{enc}}$	Frozen model's latent embedding map and exported feature width.
$k_A, k_Z; h_A, h_Z$	Structural/latent Gaussian kernels and their bandwidths.
B_A, B^l_Z, B^l_{AZ}	Trace-one input, latent, and joint Gram matrices.
$\Sigma_A, \Sigma_Z, \Sigma_{AZ}$	Corresponding marginal and joint feature-space density operators.
$S(B), \nu_j; I^l_{\mathrm{sp}}$	Spectral entropy, eigenvalues, and layer input and latent embedding information.
$b, W; J_{\mathrm{blk}}$	Spectral record budget, block length, and number of blocks.
d_h, r	Hidden width and Pearson correlation.
$X_{\mathrm{obs}}, H_e; P, Q$	Attribute/latent feature pair; joint and product-marginal laws.
F_θ, Θ, M_F	Critic, parameter class, and fixed output bound.
$J_\theta, \widehat{J}_\theta$	Population and empirical NWJ objectives.
$\varepsilon_{\mathrm{app,est,opt}}$	Approximation, estimation, and optimization errors.
$H^{l,t}_A; H^{l,t}_X$	Preceding random states before structural/feature sampling.
$\mathcal{U}, \mathcal{C}_t; \phi, p_0$	Potential/candidate edge sets; posterior/prior gate probabilities.
$\widetilde{M}, \tau_r; m, s, \xi$	Relaxed gates and temperature; feature mean, scale, and noise.
$k_{\mathrm{nbr}}, \Delta t$	Encoder neighbor budget and look-back horizon in windows.
$\Delta_{\mathrm{win}}, \delta_i(t)$	Snapshot duration and time since node i's previous event update.
$B_{\mathrm{BFS}}, N_-, N_+$	BFS node budget and sampled negative/positive edge counts.
$G' = Y, q_\psi; \lambda, \beta$	Task labels, predictor; metric and compression weights.
$D = (A, X); C_Y$	Observed input structure/features and target class count.
$\mathcal{Q}, m, B, \delta$	Readout class, sample count, loss bound, and failure probability.
$\mathcal{R}_{\log}, \alpha_{\mathcal{Q}}$	Population log-risk and readout approximation error.
$\mathrm{Rad}_m, \epsilon, \eta_{\mathrm{opt}}$	Expected Rademacher complexity, probability floor, and readout optimization tolerance.