Atomic Motion Awareness for Hydrogen Bonds Prediction in Dynamic Molecular Graphs

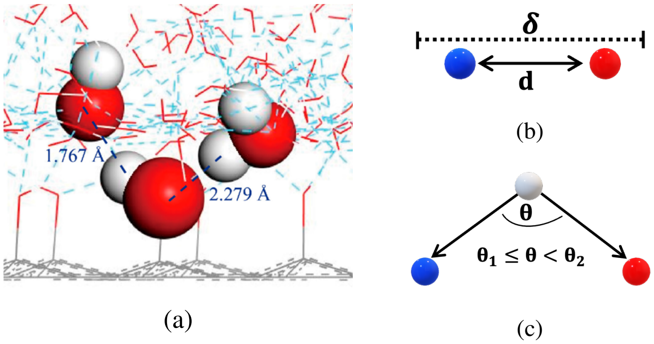
Md Hasan Anowar
Dept. of ECE
Iowa State University
Ames, Iowa, USA
ORCID: 0000-0001-7697-474X
Email: mhanowar@iastate.edu

Goce Trajcevski
Dept. of ECE
Iowa State University
Ames, Iowa, USA
ORCID: 0000-0002-8839-6278
Email: gocet25@iastate.edu

Ashfaq Khokhar
Dept. of ECE
Iowa State University
Ames, Iowa, USA
ORCID: 0000-0002-6504-8502
Email: ashfaq@iastate.edu

***Abstract*—Predicting atomic interactions that lead to hydrogen bond (HB) formation is fundamental for understanding biochemical processes and accelerating drug discovery. When dealing with such interactions, existing approaches ignore the temporal evolution of the atoms; however, in dynamic molecular systems atoms move continuously in 3D space. In such settings, molecular spatial configurations vary, and HBs arise only when certain geometric constraints co-occur. We propose a novel dual link prediction (DLP) framework that predicts the co-occurrence of geometric constraints enabling HB formation from dynamic molecular graphs. DLP employs an edge encoder to fuse the atomic relationships and temporal dynamics, a transformer-based encoder to learn node representations, and an effective memory module. Experiments demonstrate that DLP consistently outperforms state-of-the-art baselines in both prediction accuracy and computational efficiency.**

***Index Terms*—Atomic Motion, Dynamic Molecular Graph, Link Prediction**



(a)

(b)

(c)

Fig. 1: Structural formations from molecular motion: (a) snapshot of atomic movement [7] (b) two atoms in proximity ($d \leq \delta$) (c) three atoms satisfying angular range ($\theta_1 \leq \theta < \theta_2$).

I. Introduction

Technological advancements in computational drug discovery have generated large amounts of data on atomic and molecular interactions. Traditional machine learning approaches struggle to handle such large datasets efficiently, as they often depend on complex and time-consuming preprocessing steps [1]. To address this limitation, Graph Neural Networks (GNNs) have emerged as a powerful framework for learning molecular representations and predicting chemical properties [2], [3]. But most existing molecular GNNs operate on static graphs in which atoms and bonds are assumed fixed, and they typically target global molecular property prediction [4]. However, in reality, atoms as part of the molecules move continuously due to thermal fluctuations and intermolecular forces. This continuous motion leads to chemical interactions to evolve over time. Static GNNs can not capture these temporal dynamics, which are essential for modeling dynamic molecular phenomena such as bond formation and breaking. Although dynamic GNNs have shown success in domains such as transportation networks [5] and recommender systems [6], their application to modeling dynamic molecular graphs remains limited.

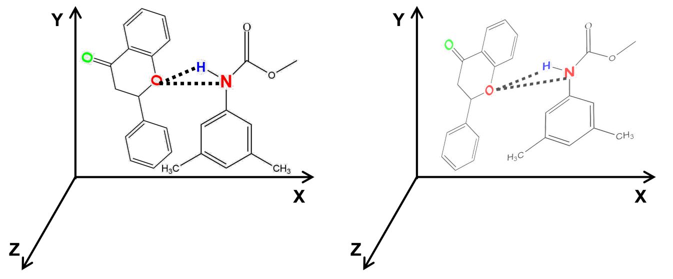
To reduce costs and speed up drug development, researchers often resort to molecular dynamics simulations (MDS) [8] to generate atomic motion data. Fig.1a shows a snapshot of a group of moving atoms at a given time. Many chemically meaningful events are defined by the simultaneous satisfaction of multiple geometric conditions, which vary due to atomic motions. For instance, a hydrogen bond (HB) is typically declared present only when (i) two atoms are within a distance threshold and (ii) the corresponding bond angle lies within a prescribed range (details in Sec. II-B). Fig.1b and Fig.1c illustrate distance-based and angle-based conditions on the same underlying system shown in Fig.1a. Treating distance-based and angle-based events separately using existing methods can misclassify HBs when only one condition is satisfied.

The distance-type and angle-type interactions arise from the physical motion of atoms and therefore evolve continuously over time. As atoms move through 3D space, these geometric relations can appear, disappear, or reoccur, reflecting an inherent form of fine-grained mobility at the atomic level. Additionally, instead of predicting distance-type and angle-type interactions independently, we focus on the joint event in which both geometric conditions are satisfied simultaneously. Designing a model for this task is challenging for three reasons. First, the two types of links are geometric, not purely topological: the model must consider chemistry-aware constraints beyond graph connectivity. Second, MDS data are long and irregular, with bursty activity and gaps that make it difficult to infer long-range dependencies from a short temporal window. Third, HB formations are chemically constrained: only specific atoms from certain molecules can participate, so treating all geometrically plausible pairs as candidates leads to many false positives.

Existing temporal graph models rely on sequential encoders or auxiliary memory modules to capture temporal dependencies [9]–[12]. While effective for single link prediction, these models are not well-suited to our setting, where prediction depends on two geometry-dictated (distance and angle-type) interactions that must be scored as a joint event. This mismatch is amplified under sparse and irregular timesteps: a windowed encoder often observes only a limited fragment of history and may fail to retain stable, node-specific regularities across long gaps. Recent work partially addresses long-horizon dependence by coupling encoders with per-node memories that are updated at every event [13], [14]. However, updating on every event policy is suboptimal for dual link prediction, as a distance-type event often occurs without a corresponding angle-type event at the same timestamp (and vice versa). Indiscriminately updating memory on such a one-sided event increases computational and storage cost and, more importantly, pollutes the stored state with interactions that are not valid evidence for *dual link* events.

In this work, we propose a novel dual link prediction (DLP) framework for detecting HB formation in dynamic molecular graphs. Here, the concept of *dual link* is essential to capture the simultaneous coexistence of distance-based and angle-based events. DLP utilizes a transformer encoder with patching to summarize each node's recent interaction history: self-attention captures dependencies within a fixed temporal window, while patching compresses long histories into locally coherent representations. Now, HB formation often exhibits node-specific *preferences* (an atom repeatedly interacts with a chemically similar atom) over longer horizons than a single window can cover. To capture this phenomenon, we introduce a lightweight preference memory that maintains an exponential moving average of recent destinations for each node and edge type, updating only when a dual-link event occurs. A learned gate controls how strongly this preference memory influences the dual link score, enabling the model to rely on it only when dual history is informative. Additionally, HB formation involves specific covalently bonded atoms rather than arbitrary atom combinations. This chemical constraint motivates explicitly encoding valid covalent identities in the edge representation. Therefore, we extend the edge encoder with a *covalent group encoding* module that integrates chemically valid pairing constraints with temporal information, reducing geometrically plausible but chemically infeasible predictions. To our knowledge, this is the first attempt to incorporate the evolution of the atoms' locations over time to predict HB formation. Our key contributions are summarized as follows:

- We introduce *dual links* to model HB as the coexistence of distance-based and angle-based interactions in dynamic molecular graphs.



(a) (b)

Fig. 2: Example temporal graph: Atoms change locations – but only the dotted lines corresponding to potential HB interactions are considered as temporal edges.

- We propose a DLP model for predicting dual links, which combines an edge encoder, a patched Transformer, and a node preference memory module. We design the *Node Preference Memory* module as edge-type separated and updated only on dual link events. Additionally, we extend the edge encoder to fuse temporal information with *covalent group encoding*, which encodes chemistry-aware constraints into edge features.
- We evaluate DLP on three real-world MDS datasets and show improved predictive accuracy and runtime efficiency over strong baselines, with modest memory overhead.

The rest of the paper is organized as follows: we introduce the preliminaries and problem definition in Sec. II. Sec. III presents the proposed DLP model, algorithm and complexity analysis. Sec. IV elaborates on the datasets, and experimental results. Sec. V provides a literature review, and Sec. VI gives concluding remarks and outlines directions for future work.

II. Preliminaries and Problem Formulation

In this section, we first introduce the preliminaries, including the definitions of temporal graphs and HB. Subsequently, we elaborate on how dynamic atomic interactions can be represented as graph structures and formally introduce the concept of *dual link* and the criteria for detecting its existence. Finally, we present the precise problem that this paper addresses, namely, the dual link prediction task.

A. Graphs

A graph is typically represented as $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of vertices (or nodes), and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of edges (or links). When dealing with dynamic scenarios in which the features of entities, including locations, evolve over time, the concept of a temporal graph naturally emerges. The temporal graph formulation has already been effectively utilized in the literature, and we have the following definition based on these works [9], [15]:

Def. 1 (Temporal Graph): A temporal graph is defined as $\mathcal{G}(t) = (\mathcal{N}, \mathcal{E}(t))$, where $\mathcal{N}$ represents the set of all vertices,

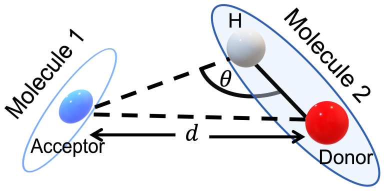


Fig. 3: An example of (criteria for) HB Formation [20]

and $\mathcal{E}(t)$ is the set of all the temporal edges observed in any time instant $\leq t$.

Here, each element $e_{ij}(t_m) \in \mathcal{E}(t)$ represents an edge between node n_i and node n_j at timestamp t_m with $t_m \leq t$. We consider the evolution of a molecular system composed of a finite set of atoms, and model it as a temporal graph $\mathcal{G}(t) = (\mathcal{N}, \mathcal{E}(t))$, where each node $n_i \in \mathcal{N}$ corresponds to a unique atom and $\mathcal{E}(t)$ captures time-stamped connections among atoms up to time t. Fig. 2 illustrates two snapshots of the same pair of molecules. The solid black lines represent each molecule's intramolecular connectivity, i.e., permanent structure, and remain unchanged throughout the interactions in the system; these corresponds to covalent bonds [16]. In contrast, molecules move relative to one another, so proximity and angular configurations may transiently satisfy HB criteria. These interaction-driven contacts are represented by dotted lines in Fig. 2, which may appear or disappear over time. We focus on these dynamic structures, i.e., we treat the dotted, geometry-induced interactions between two molecules as temporal edges in $\mathcal{E}(t)$, rather than the permanent intramolecular bonds. Thus, the evolution of $\mathcal{E}(t)$ specifically refers to the formation and breaking of such geometry-induced intermolecular interactions, while covalent bonds remain fixed. Formally, each atom in the system corresponds to a node $n_i \in \mathcal{N}$ with attributes:

$$\mathtt{attr}(\mathtt{n_i}, \mathtt{t}) = \big(\mathtt{mol}(\mathtt{n_i}), \mathtt{atype}(\mathtt{n_i}), \mathtt{r_{n_i}}(\mathtt{t})\big), \qquad (1)$$

where $mol(n_i)$ identifies the molecule containing the atom, $atype(n_i)$ is the atom type, and $r_{n_i}(t) \in \mathbb{R}^3$ is the atom's 3D position at time t. While $\mathcal{N}$ remains fixed, the edge set $\mathcal{E}(t)$ evolves over timestamps $t_1 < \cdots < t_T$ as atomic and molecular motion induces HB formation and disappearance. At each timestamp t_m, we add an edge $e_{ij}(t_m) \in \mathcal{E}(t)$ whenever the relevant chemistry-derived geometric criteria, i.e., HB distance and angle thresholds, are satisfied.

B. Hydrogen Bond (HB)

In this section, we discuss HB. HB is a type of chemical bond typically formed between a hydrogen (H) atom covalently bonded to a highly electronegative atom (called *donor*) such as nitrogen (N) or oxygen (O) interacts with a second atom that has a lone pairs of electrons (called *acceptor*), such as another N, O, or fluorine (F) [17], [18]. Fig. 3 shows an example of an HB formation. As HB formation is crucial in many biochemical and physical processes, researchers are interested in their formation (and persistence) [19].

C. Transformed Representations

The process for transforming graph structure representations consists of defining two types of temporal edges: distance-type and angle-type, which explicitly reflect the geometric and structural conditions for HB formation. We partition HB-relevant atoms into three disjoint sets using attributes $atype(n_i)$ and $mol(n_i)$: $U = \{n_i \in \mathcal{N} \mid n_i$ is a node corresponding to an acceptor atom$\}$, $V = \{n_i \in \mathcal{N} \mid n_i$ is a node corresponding to a donor atom$\}$, and $W = \{n_i \in \mathcal{N} \mid n_i$ is a node corresponding to an H atom$\}$. By construction, $U \cap V = U \cap W = V \cap W = \varnothing$ and $U \cup V \cup W \subseteq \mathcal{N}$ (in many practical situations, $U \cup V \cup W \subset \mathcal{N}$). We focus on a subset of $\mathcal{E}(t)$ involving nodes in U, V, and W, since only *acceptor–donor–H* combinations can participate in HB formation. Accordingly, we construct distance-type edges over pairs $(u, v) \in U \times V$ and angle-type edges over triplets $(u, w, v) \in U \times W \times V$, while all other interactions outside these sets are excluded from further processing.

Recall that the HB formation requires two geometric constraints – specifically: (i) the *donor–acceptor* distance (d) must fall within a threshold δ, typically within 2.2-4.0Å, and (ii) the angle for *acceptor-H-donor* triplet (θ) must lie within an angular range $[\theta_1, \theta_2)$ commonly 135°-180° [21]. Let us illustrate the geometric thresholds necessary for HB existence with an example from a real MDS dataset `flavanone255k` (cf. Sec. IV-A). For this dataset, Molecule 1 is *Flavanone* and Molecule 2 is *Polymer*, *acceptor* is an O atom, and *donor* is a N atom. The O atom comes from the *Flavanone*, and the N and H atoms belong to the *Polymer*. Following domain guidance, we use a distance threshold of 3.5Å. Thus, for this example, $d(O, N) \leq 3.5$Å and $135^\circ \leq \angle OHN < 180^\circ$ must hold.

Although these geometric rules define HB existence at an instant, molecular systems evolve continuously due to thermal motion and intermolecular forces. Consequently, *donor–acceptor* distances and bond angles evolve over time. The configuration that qualifies as an HB at time t may break at $t + 1$ and, possibly, reform later. To explicitly model this dynamic behavior, we represent the system as a temporal graph, where each satisfied geometric constraint generates a time-stamped edge of the corresponding type. Here, we formalize the HB conditions as two types of edges:

(a) **Distance edges.** A distance-type edge $e^{\text{dist}}_{u,v}(t)$ between nodes $u \in U$ and $v \in V$ at time t exists if their Euclidean distance, $d_{u,v}(t)$, satisfies:

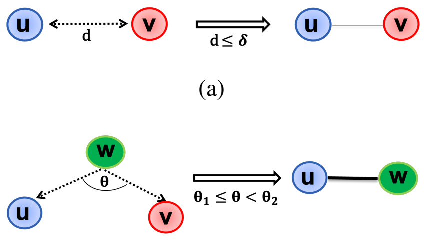
$$\mathtt{d_{u,v}}(\mathtt{t}) = \|\mathtt{r_u}(\mathtt{t}) - \mathtt{r_v}(\mathtt{t})\|_2 \leq \delta. \qquad (2)$$

(b) **Angle edges.** An angle-type edge $e^{\text{angle}}_{u,w}(t)$ between nodes $u \in U$ and $w \in W$ at time t exists if:

$$\theta_1 \leq \theta_{\mathtt{u,w,v}}(\mathtt{t}) < \theta_2, \qquad (3)$$

where $\theta_{\mathtt{u,w,v}}(\mathtt{t}) = \arccos\left(\frac{(\mathtt{r_u}(\mathtt{t}) - \mathtt{r_w}(\mathtt{t})) \cdot (\mathtt{r_v}(\mathtt{t}) - \mathtt{r_w}(\mathtt{t}))}{\|\mathtt{r_u}(\mathtt{t}) - \mathtt{r_w}(\mathtt{t})\| \, \|\mathtt{r_v}(\mathtt{t}) - \mathtt{r_w}(\mathtt{t})\|}\right)$.

These definitions yield a structured temporal graph (Fig. 4), with distance-type edges capturing proximity between atom pairs and angle-type edges corresponding to angular relationships among atom triplets.



(a)

(b)

Fig. 4: Graphical illustration of the transformed representation: (a) distance-type edge $e^{\text{dist}}_{u,v}(t)$ (b) angle-type edge $e^{\text{angle}}_{u,w}(t)$.

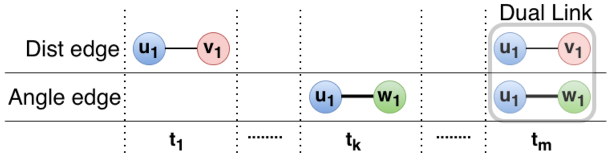


Fig. 5: Existence of dual link at time t_m (v_1 and w_1 belong to the same molecule).

Temporal Dynamics: As atomic positions $\mathtt{r}_{n_i}(t)$ evolve over time due to atomic motion in MDS, both $d_{u,v}(t)$ and $\theta_{u,w,v}(t)$ are dynamic functions of t. Thus, the existence of distance-type and angle-type edges is not static but fluctuates with atomic trajectories. A valid HB corresponds to the simultaneous satisfaction of both constraints, which we refer to as a *dual link*. Formally, we define the concept of *dual link* existence as follows:

Def. 2 (Dual Link): Given nodes $u \in U, v \in V, w \in W$, a *dual link*, $e^{\text{dual}}_{u,v,w}(t)$, exists at time t if both the corresponding distance-type $e^{\text{dist}}_{u,v}(t)$ and angle-type edges $e^{\text{angle}}_{u,w}(t)$ are present at time t.

Example: Fig. 5 illustrates interactions between nodes across different timestamps. At time $t = t_1$, node u_1 establishes a distance-type edge with node v_1, but does not form any angle-type edge. At time $t = t_k$, node u_1 creates an angle-type edge with node w_1. At t_1 and t_k, *dual link* does not exist, as they have only one kind of edge. Now, we observe a dual link at time $t = t_m$. Here node u_1 simultaneously forms both edge types, distance-type with node v_1 and angle-type with node w_1. Thus, the pair of edges $\{e^{\text{dist}}_{u_1,v_1}(t_m),\ e^{\text{angle}}_{u_1,w_1}(t_m)\}$ exemplifies the existence of a dual link.

Problem Statement: Given a temporal molecular graph $\mathcal{G}(t)$, the task is to predict the existence of a dual link, $e^{\text{dual}}_{u,v,w}(t)$, at a future time t_n using all historical information up to t_{n-1} for a candidate triplet (u, v, w). In contrast to static link prediction, the model must capture how distance and angle-based interactions evolve over time: *dual link* emerges as a consequence of continuous atomic motion rather than independent, memoryless events. Accurately predicting dual links thus amounts to forecasting HB formation, which is important in applications such as drug development and molecular modeling.

III. Methodology

In this section, we present the proposed DLP framework for dynamic molecular graphs. DLP consists of three main components: an edge encoder, a transformer encoder with patching, and a node preference memory module. The overall architecture is shown in Fig. 6a, and the memory update mechanism is illustrated in Fig. 6b. The edge encoder combines temporal information with covalent group identity features at the edge level. The transformer encoder summarizes each node's interaction history within a fixed window. The node preference memory stores per-node, per-edge-type exponential moving averages of recent positive destinations and is updated only when a dual link occurs. Finally, the dual-link predictor combines the learned node representations and the memory bias to score the existence of dual links. We detail each component in the following subsections.

We first construct historical interaction sequences for both distance-type and angle-type edges by collecting each node's past interactions as time-ordered tuples. These interactions are sorted by interaction time to retain recency. When multiple links occur at the same timestamp, each interaction is treated as a separate entry in the sequence. Fig. 7a and 7b illustrate the historical interactions and resulting sequence for node u_1. Each sequence captures local structural changes and the evolving interaction context around the node. These temporally ordered sequences are then fed into the edge encoder.

A. Edge Encoder Module

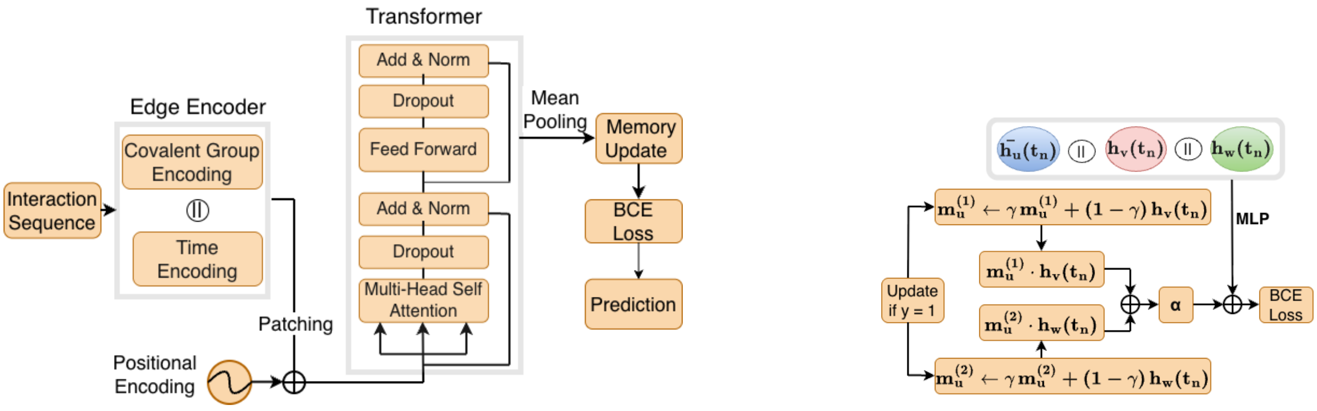
The edge encoder module consists of two submodules: temporal encoding and covalent group encoding. By integrating covalent group identity and time encoding in the edge encoder, we fuse the atomic structure relationship and temporal dynamics into the model.

1) Temporal Encoding: In a temporal heterogeneous network, utilizing the timestamps for the link formation is crucial for modeling evolving network structures. We incorporate the temporal encoding to explicitly model the evolution of atomic interactions over time. The idea behind the technique is that interactions occurring within similar timeframes exhibit similarities in their encoding.

$$\psi(\mathtt{t_n} - \mathtt{t}) = \cos\left((\mathtt{t_n} - \mathtt{t}) \times \omega\right) \tag{4}$$

where $\omega = \left(\alpha^{-\frac{\mathtt{i}-1}{\beta}}\right)_{\mathtt{i}=1}^{\mathtt{d_t}}$ is a fixed d_t dimensional vector, and $\alpha = \beta = \sqrt{d_t}$. As ω is not updated during the training phase, it simplifies the model's optimization and leads to performance improvements [15]. This technique preserves similarity among interactions occurring close in time to obtain temporal dynamics that align with other temporal graph learning techniques.

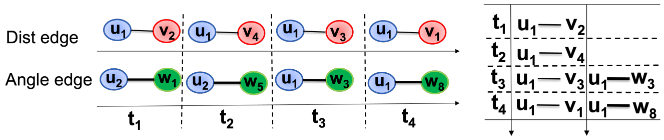
2) Covalent Group Encoding: According to the HB definition in Sec. II-B, every *H* atom is covalently bonded to a specific *donor* atom within the same molecule. Consequently, valid HB interactions must respect these fixed *donor–H* covalent groups rather than considering arbitrary combinations. Encoding this structural constraint into the representation is


Transformer
Add & Norm
Dropout
Feed Forward
Add & Norm
Dropout
Multi-Head Self Attention
Mean Pooling
Memory Update
BCE Loss
Prediction
Edge Encoder
Covalent Group Encoding
Time Encoding
Interaction Sequence
Patching
Positional Encoding
$\bar{h}_u(t_n)$ || $h_v(t_n)$ || $h_w(t_n)$
$m_u^{(1)} \leftarrow \gamma m_u^{(1)} + (1-\gamma) h_v(t_n)$
MLP
$m_u^{(1)} \cdot h_v(t_n)$
Update if y = 1
α
BCE Loss
$m_u^{(2)} \cdot h_w(t_n)$
$m_u^{(2)} \leftarrow \gamma m_u^{(2)} + (1-\gamma) h_w(t_n)$

(a)

(b)

Fig. 6: (a) Overview of the DLP architecture. (b) Memory update mechanism. Each memory maintains a compact representation of long-term preferences by summarizing the recent destinations.



(a)

(b)

Fig. 7: (a) Historical interactions of node u_1, (b) Interaction sequence for node u_1 before time t_5.

therefore essential for capturing chemically valid interactions and reducing false-positive predictions.

Recall from Sec. II-C, the nodes corresponding to the *donor* and H atoms are partitioned into two disjoint node sets V and W, respectively, where $V, W \subseteq \mathcal{N}$. To identify the covalent groups, we determine the donor node (representing the covalently bonded *donor* atom) associated with each H node using a distance-based criterion at an initial reference time. Specifically, for each $w \in W$, we select the $v \in V$ that minimizes the Euclidean distance $\arg\min \|\mathbf{r}_v(0) - \mathbf{r}_w(0)\|_2$, where $\mathbf{r}_v(0)$ and $\mathbf{r}_w(0)$ denote the 3D coordinates of nodes v and w at time $t = 0$. As covalent bonds are fixed, these covalent groups remain constant across all timestamps.

Each identified covalent group is then assigned a discrete categorical index. We encode the identity using a one-hot vector $\mathbf{x}^{\text{group}} \in \mathbb{R}^C$, as one-hot encoding is a simple and computationally efficient method. C denotes the number of unique covalent groups and $\mathbf{x}^{\text{group}} = [0, \ldots, 1, \ldots, 0]$, is a binary vector with a single nonzero entry corresponding to the covalent group index. This covalent group encoding injects chemically valid pairing constraints into the edge representation, ensuring that the model distinguishes structurally feasible atomic configurations from geometrically plausible but chemically infeasible ones.

To construct the dual-edge representation, we combine temporal encoding, covalent group encoding, and raw edge features. For a distance-type edge, we fuse $\psi_{u,v}(t_n - t)$, $x^{\text{group}}_{u,v}$, and $x^{\text{edge}}(e^{\text{dist}}_{u,v}(t))$. For an angle-type edge, we combine $\psi_{u,w}(t_n - t)$, $x^{\text{group}}_{u,w}$, and $x^{\text{edge}}(e^{\text{angle}}_{u,w}(t))$. The two representations are concatenated and passed through a one-layer MLP to obtain the final dual edge embedding:

$$\mathtt{X}\left(\mathtt{e}^{\text{dual}}_{\mathtt{u,v,w}}(\mathtt{t})\right) = \text{MLP}\Big(\underbrace{\psi_{\mathtt{u,v}}(\mathtt{t_n} - \mathtt{t}) \parallel \mathtt{x}^{\text{group}}_{\mathtt{u,v}} \parallel \mathtt{x}^{\text{edge}}\left(\mathtt{e}^{\text{dist}}_{\mathtt{u,v}}(\mathtt{t})\right)}_{\text{distance-type}} \parallel \underbrace{\psi_{\mathtt{u,w}}(\mathtt{t_n} - \mathtt{t}) \parallel \mathtt{x}^{\text{group}}_{\mathtt{u,w}} \parallel \mathtt{x}^{\text{edge}}\left(\mathtt{e}^{\text{angle}}_{\mathtt{u,w}}(\mathtt{t})\right)}_{\text{angle-type}}\Big). \tag{5}$$

B. Transformer Encoder Module with Patching

Traditional sequence models often struggle when handling lengthy interaction sequences due to diminishing correlations among distant input tokens. As the nearby tokens exhibit stronger semantic relationships, we incorporated the patching method introduced in Vision Transformer (ViT) [22]. Patching technique splits the sequence into patches, emphasizing local semantic coherence. Then, it aggregates tokens at the patch level rather than individually, thereby improving the efficiency of capturing interactions by reducing the sequence length. Let us consider $S_u(t_n)$ as the interaction sequence containing all historical links involving node u, which includes all nodes that interacted with u before t_n. Since atomic interaction sequences are one-dimensional, they can be directly segmented into patches. Let $X_u(t_n) \in \mathbb{R}^{|S_u(t_n)| \times d_S}$ denotes the edge embeddings of $S_u(t_n)$. We divide this one-dimensional sequence into patches of fixed length p, and each patch has $N = |S_u(t_n)|/p$ interactions. We write $X^p_u(t_n) \in \mathbb{R}^{N \times d_P}$ for the resulting patch embeddings. Standard positional encodings $p_k \in \mathbb{R}^{d_P}$ are added to each patch to retain order information. We use the classic transformer encoder [23] to capture higher-order dependencies within temporal interaction sequences. It contains two main components: (i) a multi-head self-attention layer that computes contextualized representations by modeling dependencies across interactions, and (ii) a feed-forward network with residual connections to stabilize training. The self-attention mechanism projects each input embedding, $X_u^p(t_n)$, into queries, keys, and values via learnable weight matrices W_Q, W_K, W_V. For a sequence of embeddings, the attention scores are computed using scaled dot-product attention: $Attn(Q, K, V) = softmax\left(\frac{QK^\top}{\sqrt{d_{key}}}\right)V$, where d_{key} is the key dimension. Each interaction embedding is updated as a weighted sum of all value vectors, with weights determined by the similarity between its query and all other keys. Formally, given the sequence embedding $X_u^p(t_n)$ for node u, the output of one transformer block H_{out} is:

$$\mathtt{Q} = \mathtt{X}_\mathtt{u}^\mathtt{p}(\mathtt{t}_\mathtt{n})\,\mathtt{W}_\mathtt{Q}, \quad \mathtt{K} = \mathtt{X}_\mathtt{u}^\mathtt{p}(\mathtt{t}_\mathtt{n})\,\mathtt{W}_\mathtt{K}, \quad \mathtt{V} = \mathtt{X}_\mathtt{u}^\mathtt{p}(\mathtt{t}_\mathtt{n})\,\mathtt{W}_\mathtt{V}. \tag{6}$$

$$\begin{aligned} \mathtt{H}_\mathtt{res} &= \mathtt{X}_\mathtt{u}^\mathtt{p}(\mathtt{t}_\mathtt{n}) + \mathtt{Attn}(\mathtt{Q}, \mathtt{K}, \mathtt{V}), \\ \mathtt{H}_\mathtt{out} &= \mathrm{H}_\mathrm{res} + \mathtt{FFN}\big(\mathtt{LN}(\mathrm{H}_\mathrm{res})\big). \end{aligned} \tag{7}$$

where LN denotes layer normalization, and FFN is a two-layer feed-forward network. We employ LN before the feed-forward blocks because it provides effective optimization [24]. Stacking L such blocks yields the final patch-level representation, which we average-pool to compress H_{out} to obtain the link representation $h_u^{link}(t_n)$ for node u at time t_n.

To capture the local structural context, we aggregate the node features from the 1-hop neighborhood of u over the time window $[t, t_n]$. Let $\mathcal{P}(u; t_n, t)$ be the set of neighbors of u in this window, and let X_v denote the node features of v. The neighborhood information is aggregated as follows:

$$\mathtt{h}_\mathtt{u}^\mathtt{node}(\mathtt{t}_\mathtt{n}) = \mathtt{X}_\mathtt{u} + \mathtt{Mean}\big\{\mathtt{X}_\mathtt{v} \mid \mathtt{v} \in \mathcal{P}(\mathtt{u}; \mathtt{t}_\mathtt{n}, \mathtt{t})\big\}. \tag{8}$$

We concatenate the link representation and embedding of node u at time t_n to get the final representation, $h_u(t_n) = \big[h_u^{link}(t_n) \,\|\, h_u^{node}(t_n)\big]$. We use similar techniques to obtain $\mathtt{h}_v(t_n)$ and $\mathtt{h}_w(t_n)$ for the v and w nodes, respectively.

C. Node Preference Memory Module

In this module, we maintain a persistent vector for each source node and each edge type. For a source u, we denote $m_u^{(1)}, m_u^{(2)} \in \mathbb{R}^D$ as memory blocks for the distance and angle edge-type, respectively. Each $m_u^{(k)}$ summarizes the recent positive destinations of u and acts as a compact representation of long-term connection preferences. Memories are updated only on positive dual-link events, using an exponential moving average with a time aware half-life. Let $t_{\text{last}}(u)$ be the last time at which node u's memory was updated. So, $\Delta t = t_n - t_{\text{last}}(u)$, where Δt denotes the elapsed time since the last update of node u. Let $\gamma = \exp\big(-\ln 2 \cdot \Delta t / \tau\big)$ is the time decay factor. When a dual-link occurs, we update

$$\begin{aligned} \mathtt{m}_\mathtt{u}^{(1)} &\leftarrow \gamma\,\mathtt{m}_\mathtt{u}^{(1)} + \big(1 - \gamma\big)\,\mathtt{h}_\mathtt{v}(\mathtt{t}_\mathtt{n}), \\ \mathtt{m}_\mathtt{u}^{(2)} &\leftarrow \gamma\,\mathtt{m}_\mathtt{u}^{(2)} + \big(1 - \gamma\big)\,\mathtt{h}_\mathtt{w}(\mathtt{t}_\mathtt{n}). \end{aligned} \tag{9}$$

The half-life parameter τ defines the rate at which historical preferences decay. This exponential decay ensures that older interactions gradually lose importance while recent ones dominate the memory update. At prediction time, the preference memory contributes an edge-type-averaged dot-product bias to the dual-link score:

$$\mathtt{b}_\mathtt{pref} = \alpha\Big(\mathtt{m}_\mathtt{u}^{(1)} \cdot \mathtt{h}_\mathtt{v}(\mathtt{t}_\mathtt{n}) + \mathtt{m}_\mathtt{u}^{(2)} \cdot \mathtt{h}_\mathtt{w}(\mathtt{t}_\mathtt{n})\Big). \tag{10}$$

where $\alpha \in \mathbb{R}$ is a learned scalar gate (initialized to zero) that controls how strongly the preference signal influences the logit. If the history is uninformative, training keeps $\alpha \approx 0$, and the model effectively reduces to a transformer-only model.

Predictor: As the two temporal edge types, distance and angle, are extracted from the same molecular system, they share a global node ID space. For each time t_n, we therefore form dual-link candidates only for source nodes that appear in both edge-types. If node u's representations are $h_u^{(1)}(t_n)$ and $h_u^{(2)}(t_n)$ for distance-type and angle-type edges, respectively, we construct an aggregated embedding $\bar{h_u}(t_n) = h_u^{(1)}(t_n) + h_u^{(2)}(t_n)$. Sources observed on only one type of edge at t_n are treated as single events and are not used as dual links. Finally, we score the dual-link candidate using the source embedding, the two destination embeddings, and the preference bias:

$$\mathtt{p} = \mathtt{MLP}\big([\bar{\mathtt{h}_\mathtt{u}}(\mathtt{t}_\mathtt{n}) \,\|\, \mathtt{h}_\mathtt{v}(\mathtt{t}_\mathtt{n}) \,\|\, \mathtt{h}_\mathtt{w}(\mathtt{t}_\mathtt{n})]\big) + \mathtt{b}_\mathtt{pref}. \tag{11}$$

As we formulate dual link prediction as a binary classification problem, we select the Binary Cross-Entropy loss corresponding to the negative log-likelihood of a Bernoulli distribution [25].

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{\mathtt{N}} \sum_{\mathtt{i}=1}^{\mathtt{N}} \Big[\mathtt{y}_\mathtt{i} \log(\mathtt{p}_\mathtt{i}) + (1 - \mathtt{y}_\mathtt{i}) \log\big(1 - \mathtt{p}_\mathtt{i}\big)\Big]. \tag{12}$$

Here N denotes the total number of dual-link candidates in a mini-batch, and y_i is the ground-truth label: $y_i = 1$ when a dual link exists at time t, and $y_i = 0$ otherwise.

Algorithm 1 summarizes the workflow of the DLP method. Lines 1-5 describe the construction of edge features by combining temporal, covalent group, and raw edge features. Line 6 encodes them with a transformer to obtain contextualized node representations. Lines 7–12 update the preference memory with time decay, and Lines 13–15 perform dual link prediction and loss computation.

We provide a detailed analysis of the time and space complexities of the DLP method in the following section.

Complexity Analysis: Let $|\mathcal{N}|$ be the number of nodes and $|\mathcal{E}|$ denotes the total number of edges (distance-type or angle-type) observed across the dataset. Each edge instance constitutes one training example whose interaction history is transformed into a sequence of T patches. Training is performed using mini-batches of fixed size B, where each iteration processes B edge instances. The encoder consists of L transformer blocks with hidden dimension D.

Time Complexity: Processing a single edge instance with T patches using transformer encoder costs $O(T^2D) + O(TD^2)$, corresponding to attention and FFN layers. Since T is fixed and typically small, $O(TD^2)$ term dominates. For a batch of B edge instances and L layers, the cost per iteration is $O\big(BLTD^2\big)$. The predictor evaluates one positive candidate (if present) and a constant number K of negatives

Algorithm 1 Dual Link Prediction (DLP)

```
Require: memories $m_u^{(1)}, m_u^{(2)}$; last-update time $t_{\text{last}}(u)$; preference gate $\alpha$; half-life $\tau$; time $t_n$; dual labels $y$
Ensure: loss $\mathcal{L}$
    // Edge Feature Construction
for each edge $(u, v, t)$ do
    time_feat ← $\psi(t_n - t)$
    covalent_group_feat ← one_hot(covalent_group$(v, w)$)
    feat ← [time_feat || covalent_group_feat || raw_edge_feat]
end for
    // Transformer Encoding
$h_u(t_n), h_v(t_n), h_w(t_n) \leftarrow$ TransformerEncoder(feat)
    // Preference Memory Module
$\Delta t \leftarrow t_n - t_{\text{last}}(u)$
$\gamma \leftarrow \exp(-\ln 2 \cdot \Delta t / \tau)$
$m_u^{(1)} \leftarrow \gamma\, m_u^{(1)} + (1-\gamma) h_v(t_n)$
$m_u^{(2)} \leftarrow \gamma\, m_u^{(2)} + (1-\gamma) h_w(t_n)$
$t_{\text{last}}(u) \leftarrow t_n$
$b_{\text{pref}} \leftarrow \alpha \cdot \big(m_u^{(1)} \cdot h_v(t_n) + m_u^{(2)} \cdot h_w(t_n)\big)$
    // Dual-Link Prediction
$p \leftarrow$ MLP($[h_u(t_n) \,\|\, h_v(t_n) \,\|\, h_w(t_n)]$) $+ b_{\text{pref}}$
$\mathcal{L} \leftarrow$ BCEWithLogits$(p, y)$
return $\mathcal{L}$
```

(in our implementation, $K = 3$) for each edge instance. The two-layer MLP requires $O(D^2)$ per instance, yielding $O\big((K+1)BD^2\big) = O(BD^2)$. For the preference module, only positive dual events update the memory. Let $B^+ \leq B$ be the number of positives in a batch. The update cost is $O(B^+D) \subseteq O(BD)$, which is negligible compared to the encoder $O(BD^2)$ term.

Combining all, the running time per batch of B edge instances is $O\big(BLTD^2\big) + O\big(BD^2\big) + O\big(BD\big)$, with the transformer term $O(BLTD^2)$ dominating. Since B, T, and L are fixed hyperparameters, the run time scales linearly with the number of edge instances across epochs.

Space complexity: The transformer contributes $O(LD^2)$ space, and the predictor adds $O(D^2)$, giving $O(LD^2)$ overall. The preference module maintains two D-dimensional memories per node, yielding $O(|\mathcal{N}|D)$ additional space. For backpropagation, storing intermediate activations for a batch of B edge instances with T tokens requires $O(BTD)$. Thus, the overall space usage is $O(LD^2) + O(BTD) + O(|\mathcal{N}|D)$. The memory overhead therefore grows linearly with the number of nodes, while the dominant cost in practice is the transformer parameters and activations.

IV. Experiments

In this section, we perform comprehensive experiments to present the effectiveness of the proposed DLP method. Specifically, we aim to address the following research questions: RQ1: How does DLP perform on the dual-link prediction task compared to strong graph-based and temporal link prediction baselines? RQ2: What is the contribution of the node preference memory to predictive performance? RQ3: Does the covalent group encoding module provide additional gains by injecting chemistry-aware information into the edge representation? RQ4 - How does the computational cost of DLP compare to two-stage pipelines that predict links independently and enforce dual link constraints via post-processing?

TABLE I: Dataset Statistics

	flavanone255k	**flavanone80k**	**benzoin**
# Nodes	866	874	649
# Edges	14,644	67,914	74,668

A. Data Description

We evaluate our proposed DLP method using three real-world MDS datasets [26], [27]. MDS has become a valuable tool for domain scientists, offering a cost-effective alternative to laboratory experiments while reducing failure rates and accelerating drug development [8]. The significance of this MDS data lies in its ability to detect important chemical phenomena, specifically HB formation, which is of particular interest in a wide range of research [28]–[30]. The selected datasets contain atomic (and molecular) motion represented as quadruplets *(x, y, z, t)* where *(x, y, z)* denote the 3D spatial coordinates of an atom at time t. From this raw data, we derive two types of interactions: distance-type and angle-type edges, covering a simulation period of 1000 time units. We note that this data preprocessing step is necessary for all the models used in this paper. The total number of nodes and number of edges per dataset are summarized in Table I.

We perform a chronological train-validation-test split (70%-15%-15%), using existing links as positive examples and randomly sampled non-existent links as negative examples.

B. Configurations

We run our experiments on an HPC cluster using a single GPU node with an NVIDIA A100-SXM4 80 GB, equipped with dual AMD EPYC 7543 32-core processors. Each job is allocated 4 CPU cores and 64 GB RAM through Slurm. The software stack consists of Python 3.9.21, PyTorch 1.13, and CUDA 12.8 for GPU acceleration. We provide implementation details at https://github.com/hasananowar/DLP.

C. Baseline Models for Comparison

To provide a comparative analysis of our proposed DLP model, we compare it with the state-of-the-art methods from three distinct categories: static, dynamic, and molecular graph models. For static graph modeling, we select GraphSAGE [33], a popular model known for its effective aggregation of neighbor information to generate node embeddings. Dynamic graph models include TGAT [9], GraphMixer [15], and FreeDyG [31], which incorporate temporal information into node and edge embeddings to address evolving graph structures. We also compare DLP with the SkipGNN [32], which uses a skipping mechanism to gather information from both nearby and distant nodes in the graph. SkipGNN was initially designed for biological networks but can also be applied to molecular data to predict relationships without requiring detailed chemical features.

TABLE II: Performance Comparison

Method	flavanone255k				flavanone80k				benzoin			
	AUROC	AP	F1	Time (s)	AUROC	AP	F1	Time (s)	AUROC	AP	F1	Time (s)
TGAT [9]	95.59	89.19	76.99	317.17	97.58	94.13	75.69	1026.82	95.81	89.92	72.29	1088.61
GraphMixer [15]	95.32	88.86	73.31	434.98	97.6	94.3	74.9	1701.69	97.98	95.13	77.26	1729.05
FreeDyG [31]	95.95	89.75	75.93	2318.6	97.7	94.35	72.7	6308.7	98.81	97.09	79.79	5408.24
SkipGNN [32]	93.34	92.33	70.41	122.58	92.68	89.07	67.42	463.87	86.41	82.68	70.71	336.46
GraphSAGE [33]	90.18	88.18	78.75	2.25	87.26	86.53	74.64	2.22	88.41	87.87	75.78	2.21
DLP (Ours)	**97.94**	**95.44**	**90.38**	44.56	**99.4**	**98.62**	**96.66**	190.76	**99.78**	**99.48**	**98.1**	247.86

Best results are in bold. Values are reported as mean over 5 runs.

We note that existing baseline models primarily focus on single-link predictions rather than simultaneously predicting dual links. Consequently, we adapt these models to align with our *dual link* prediction scenario. Specifically, we implement each baseline separately on distance-type and angle-type edges, then combine their predictions in post-processing steps to enable dual link prediction performance evaluation. This adaptation allows for a fair comparison and highlights the effectiveness of explicitly modeling dual links.

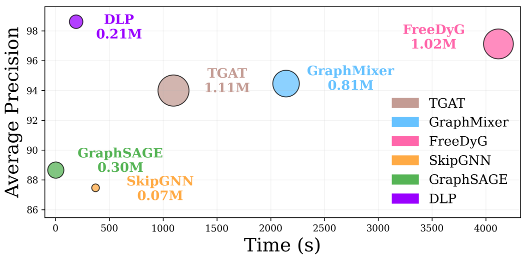


Fig. 8: Model size, performance, and run time comparison.

D. Result Analysis

Table II presents the results of a performance comparison of the proposed DLP method against the baseline models. We use AUROC, Average Precision (AP), and F1 scores as primary evaluation metrics, along with computational runtime to assess efficiency. We report performance averaged over 5 independent runs with different random seeds to assess result consistency. For each run, we train the model from scratch while keeping the data split and hyperparameters fixed, and report the mean of the evaluation metrics. Our results demonstrate that DLP consistently achieves superior predictive performance compared to all baseline models. For instance, on the `flavanone255k` dataset, DLP attains an AUROC of 97.94%, outperforming strong temporal baselines such as TGAT [9] (95.59%), GraphMixer [15] (95.32%), and FreeDyG [31] (95.95%). On `benzoin`, DLP achieves near-perfect performance (99.78% AUROC, 99.48% AP), indicating that the model generalizes well to different chemical systems. Static graph models (GraphSAGE, SkipGNN) are competitive only on the smallest dataset but underperform on the larger datasets.

An important observation is that GraphSAGE consistently shows the lowest runtime. However, there is a specific reason: we applied this model only to the graph structure at the final timestamp, given that it is a static graph model. Thus, in reality, GraphSAGE cannot capture the temporal evolution of the graph, as reflected in its performance; for example, AUROC drops to 87.26% on the `flavanone80k` dataset.

Unlike all baselines which operate on a single edge type, DLP considers distance and angle edges jointly at each timestamp. This avoids false positives arising from one-sided events (e.g., distance satisfied but angle not satisfied), which are common in molecular simulations. The higher F1 scores for DLP across all datasets indicate that joint modeling is crucial for identifying the dual links. For example, on `flavanone80k`, DLP achieves an F1 of 96.66%, whereas the best baseline (TGAT) reaches only 75.69%, despite having comparable AUROC. This reflects DLP's sharper ability to distinguish true dual links from abundant near-miss negatives.

Fig. 8 relates parameter size, AP, and training (with validation) time on `flavanone80k`. Although DLP has fewer trainable parameters (0.21M) than TGAT (1.11M), GraphMixer (0.81M), and FreeDyG (1.02M), it achieves the highest AP among all models. DLP's node preference memory introduces only a small number of non-trainable buffers, which provide temporal state without inflating model complexity and enables high accuracy under long, irregular interaction sequences where attention-based temporal models tend to struggle.

Another critical advantage of DLP is that it directly predicts dual links in a single forward pass. Baseline methods, designed for single-edge prediction, must run twice (once per edge type) and combine the outputs via additional post-processing. Fig. 9 reports total inference time, including post-processing for the baselines. DLP achieves substantial speedups relative to temporal GNNs. For example, on `flavanone255k`, DLP completes inference in 0.084 s, whereas TGAT and GraphMixer require 0.480 s and 1.132 s, respectively. On `benzoin`, FreeDyG requires 13.170 s, while DLP completes inference in 0.426 s. These results highlight that DLP architecture is not only more accurate but also computationally leaner, as it eliminates redundant passes and post-processing.

E. Hyperparameter Tuning

We study two standard encoding strategies for the covalent group feature: (i) one-hot encodings: which are parameter-free; removes the optimization and memory overhead; (ii) trainable

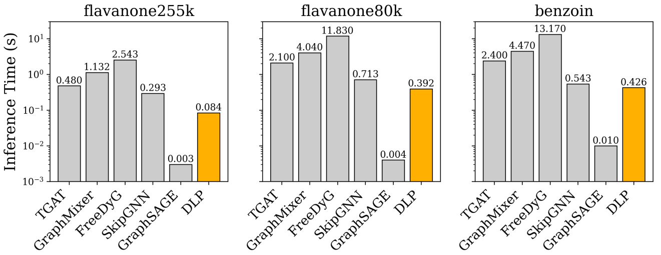


Fig. 9: Inference time comparison.

TABLE III: Covalent Group Encoding: One-hot vs. Trainable Embedding

Method	Dataset	AUROC	AP	F1	Time (s)	Memory (MB)
One-hot	flavanone255k	97.94	95.44	90.38	44.56	1.39
	flavanone80k	99.4	98.62	96.66	190.76	1.39
	benzoin	99.78	99.48	98.1	247.86	1.20
Trainable	flavanone255k	97.65	95.37	89.92	86.53	1.47
	flavanone80k	99.66	99.38	97.52	363.75	2.29
	benzoin	99.85	99.70	98.57	423.60	2.20

Reported the mean values over 5 runs.

embeddings: which introduce additional learnable parameters and can capture latent similarity among categories [34]. Our goal is to quantify whether learnable embeddings provide benefits for the *dual link* prediction task relative to their added cost. Such comparative evaluations are commonly used in literature to choose the embedding strategy from both performance and practical viewpoints [25].

Table III reports predictive performance (AUROC/AP/F1), model training time, and memory usage (parameters + buffers). For trainable embeddings, we tune the embedding dimension over $\{16, 32, 64\}$ and report results for the best configuration selected by AUROC. Overall, both variants yield comparable accuracy: trainable embeddings provide a small gain on `flavanone80k` and `benzoin`, while one-hot is slightly better on `flavanone255k`. These differences are modest, suggesting that covalent group identity is already discriminative, and introducing learnable embeddings does not consistently improve performance. In contrast, efficiency differences are substantial. Trainable embeddings increase training time by approximately 1.7–1.9 times across datasets and nearly double the memory usage (e.g., 1.39MB → 2.29MB on `flavanone80k`). Given the minimal accuracy gains and significantly higher computational cost associated with learnable embeddings, we adopt one-hot encoding as the default representation in our model.

F. Ablation Study

We now present the individual contributions of the preference memory module and covalent group encoding in the proposed DLP model.

1) Memory Module: We compare the proposed DLP model with a variant that is architecturally identical but does not use the node preference memory. We observe from Table IV that removing the memory leads to a consistent drop across all three datasets (up to 7 points in AP and more than 6 points in F1 on `flavanone255k`), confirming that the preference memory is necessary for accurately detecting dual-links.

We observe from the complexity analysis in Sec. III-C that the preference module adds only linear space in terms of the number of nodes and linear time in terms of number of edges. The preference memory does not introduce additional trainable parameters beyond a single gating scalar α, making it a lightweight extension. The only additional state it maintains is a set of non-trainable, per-node buffers. So, while the number of trainable parameters remains almost unchanged, the total memory usage by parameters and buffers slightly increases. To make a fair comparison, we report the total memory (parameters + buffers) for both variants. The model without preference memory uses $0.82\,\text{MB}$, whereas enabling the memory increases this up to $1.39\,\text{MB}$. In other words, the extra buffers account for a small fraction of the overall persistent memory, yet they yield a substantial performance improvement. This reinforces our design choice: the memory trades a modest, predictable increase in buffer space for a large performance gain.

2) Covalent Group Encoding: We compare DLP with a variant that excludes the covalent group encoding from the edge representation. As shown in Table IV, adding covalent group encoding improves accuracy and efficiency as it reduces false positives and shrinks the effective search space. Compared to DLP w/o covalent group, it yields AUROC gains of 0.2–0.7 points, AP gains of 0.5–2.5 points, and F1 improvements of 1.7–6 points. Additionally, runtime decreases by roughly a factor of 2 on all datasets (e.g., from $389.45\,\text{s}$ to $190.76\,\text{s}$ on `flavanone80k`).

V. Related Work

Link prediction on static graphs has long been approached using local heuristics, such as common neighbors and Adamic-Adar, which use handcrafted proximity scores for node pairs [35]. These methods are simple and interpretable, but are difficult to extend to temporal, heterogeneous, or geometry-aware settings and cannot encode domain-specific geometric constraints. Modern GNN-based approaches generalize this line by learning node representations and then scoring candidate links. Early architectures such as GCN [36] and GraphSAGE [33] learn node embeddings that can be combined to form edge scores, but treat links implicitly and have been shown to struggle when used for link prediction [37]. To address this, pair-aware GNNs have been proposed, including subgraph encoders such as SEAL [38], path-reasoning models such as NBFNet [39], and transformer-style decoders such as LPFormer [40]. These methods encode enclosing structures or multi-hop paths around a candidate pair [35]. They improve accuracy but often at a substantial computational cost as each candidate pair requires a separate neighborhood extraction.

Our work is closest in spirit to these pair-aware methods, but these techniques treat a link as a purely *topological* object: the representation is learned on the basis of graph connectivity.

TABLE IV: Ablation study.

Method	flavanone255k				flavanone80k				benzoin			
	AUROC	AP	F1	Time (s)	AUROC	AP	F1	Time (s)	AUROC	AP	F1	Time (s)
DLP	97.94	95.44	90.38	44.56	99.4	98.62	96.66	190.76	99.78	99.48	98.1	247.86
DLP (w/o Memory)	96.52	88.96	83.99	48.7	98.38	94.53	89.53	157.19	98.77	95.45	92.0	202.39
DLP (w/o Covalent Group)	97.45	94.05	88.67	63.0	99.2	98.13	94.23	389.45	99.05	96.99	92.05	429.69

Reported the mean values over 5 runs.

In contrast, our framework considers geometric information alongside the topological structure. In our settings, the model must predict whether a distance-type edge and an angle-type edge occur simultaneously. The coupling between the two edges is induced by shared molecular geometry (distance and angle constraints), not by topological overlap in a single graph. Classical pair encoders do not enforce such dual geometric conditions.

A complementary line of research models interaction streams as temporal graphs where temporal order is preserved via persistent, time-aware node states. Temporal GNNs such as TGAT [9], TGN [10], DyRep [11], and JODIE [12] update the node states upon incoming event. Recent heterogeneous extensions such as THAN [13] and HTGN [14], further integrate relation-aware encoders to model multi-typed neighborhood dynamics. While successful on recommendation and user-item settings, these models typically operate on a single edge process and update memory on every event, regardless of their semantic relevance. In contrast, we retain the notion of persistent node states but introduce two key modifications: (i) we transition from single-stream modeling to a dual-event process, predicting the co-occurrence of distance and angle-type interactions; (ii) we implement selective memory updates: node states are updated only upon a "dual-positive" event, where both geometric constraints are simultaneously satisfied. This architectural shift differentiates the proposed DLP method from existing models in both function and intent. While THAN and HTGN maintain a latent summary of recent neighborhood dynamics, DLP's memory serves as a specific *preference* signal that models long-term preferences of nodes. Furthermore, unlike HTGN, which feeds memory directly into the encoder, DLP utilizes an *opt-in* mechanism. Here, a learned gate controls how strongly the memory influence the dual-link score. Additionally, HTGN assumes periodic temporal patterns, while DLP addresses molecular dynamics which evolves as continuous physical processes.

A large body of prior work models chemical and biomedical domains, casting tasks such as drug-target interactions (DTIs), drug-drug interactions (DDIs), and protein-protein interactions (PPIs) as link prediction problems. Approaches such as DTINet [41] embed heterogeneous biological information into a low-dimensional space and infer DTIs via proximity in the embedding space. Deep sequence-based methods such as DeepDDI [42] directly learn interaction patterns from drug descriptors. Graph-based approaches, including Decagon [43] (applies multirelational GCNs to drug-drug-side-effect networks), and NeoDTI [44] (aggregates neighborhood signals across multiple biomedical association networks) exploit the relational structure of static knowledge graphs. These formulations typically predict a single edge type, and lack temporal dynamics or geometric constraints.

Recent work has also begun to address higher-order temporal structures. In this study [45], the authors study motif transition processes in temporal graphs to learn how motifs appear and evolve over time. This work is related to our motivation of capturing recurring interaction patterns, but their objective is to characterize and predict motif transitions, rather than predicting chemically defined dual events. Triangle completion time prediction (TCTP) [46] is another related problem, where the goal is to predict when the third edge of a triangle will appear in a dynamic network. However, TCTP focuses on closing topological triangles in a single temporal network, whereas our task enforces dual geometric conditions that must be satisfied simultaneously. In addition, our task differs from prior works in that we predict a dual link in a dynamic molecular system which requires encoding chemistry-aware edge features, i.e., covalent group identity, and incorporating a preference memory to capture long-horizon interaction tendencies. To our knowledge, existing topological, temporal, and biochemical methods do not model these joint geometric constraints in continuous time.

VI. Conclusion

In this paper, we introduced the notion of *dual links* in dynamic molecular graphs, characterizing HB formation through the simultaneous satisfaction of distance and angular constraints. We presented DLP, a unified framework that integrates a transformer encoder with a memory module to predict *dual links*. Our experiments demonstrate that DLP outperforms static, temporal, and biomolecular graph baselines while minimizing computational overhead. An extension of this research is to move beyond instantaneous event prediction to model HB persistence. In a molecular system, HBs frequently form, break, and reform, introducing temporal causality. Consequently, the persistence of HB depends not only on whether distance and angle thresholds are satisfied at a single timestamp but also on the temporal consistency of these conditions. Future work will investigate models capable of predicting persistence (e.g., bond stability) from the temporal evolution of *dual-link* events.

References

[1] K. Abbas, A. Abbasi, S. Dong, L. Niu, L. Yu, B. Chen, S.-M. Cai, and Q. Hasan, "Application of network link prediction in drug discovery," *BMC bioinformatics*, vol. 22, pp. 1–21, 2021.

[2] W. Tai, T. Zhong, G. Trajcevski, and F. Zhou, "Redundancy undermines the trustworthiness of self-interpretable gnns," in *International Conference on Machine Learning*, 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:283567344

[3] T. Zhao, D. Luo, X. Zhang, and S. Wang, "Towards faithful and consistent explanations for graph neural networks," *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:254854393

[4] X. Wu, H. Wang, Y. Gong, D. Fan, P. Ding, Q. Li, and Q. Qian, "Graph neural networks for molecular and materials representation," *Journal of Materials Informatics*, vol. 3, no. 2, 2023. [Online]. Available: https://www.oaepublish.com/articles/jmi.2023.10

[5] A. Ali, Y. Zhu, and M. Zakarya, "Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction," *Neural Networks*, vol. 145, pp. 233–247, 1 2022. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0893608021004123

[6] C. Gao, X. Wang, X. He, and Y. Li, "Graph neural networks for recommender system," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, ser. WSDM '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 1623–1625. [Online]. Available: https://doi.org/10.1145/3488560.3501396

[7] M. H. Anowar, A. Shamail, X. Wang, G. Trajcevski, S. Murad, C. J. Jameson, and A. Khokhar, "Compressing generalized trajectories of molecular motion for efficient detection of chemical interactions," *Information Systems*, vol. 125, p. 102426, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030643792400084X

[8] M. Aminpour, C. Montemagno, and J. A. Tuszynski, "An overview of molecular modeling for drug discovery with specific illustrative e.g's of apps." *Molecules*, vol. 24, 2019.

[9] D. Xu, C. Ruan, E. Körpeoglu, S. Kumar, and K. Achan, "Inductive representation learning on temporal graphs," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=rJeW1yHYwH

[10] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, "Temporal graph networks for deep learning on dynamic graphs," in *ICML 2020 Workshop on Graph Representation Learning*, 2020.

[11] R. S. Trivedi, M. Farajtabar, P. Biswal, and H. Zha, "Dyrep: Learning representations over dynamic graphs," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:108296188

[12] S. Kumar, X. Zhang, and J. Leskovec, "Predicting dynamic embedding trajectory in temporal interaction networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1269–1278. [Online]. Available: https://doi.org/10.1145/3292500.3330895

[13] L. Li, L. Duan, J. Wang, C. He, Z. Chen, G. Xie, S. Deng, and Z. Luo, "Memory-enhanced transformer for representation learning on temporal heterogeneous graphs," *Data Science and Engineering*, vol. 8, pp. 98–111, 6 2023.

[14] C. Yue, L. Du, and M. Chen, "Heterogeneous temporal graph neural networks for link prediction," in *Companion Proceedings of the ACM on Web Conference 2025*, ser. WWW '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 1505–1509. [Online]. Available: https://doi.org/10.1145/3701716.3715474

[15] W. Cong, S. Zhang, J. Kang, B. Yuan, H. Wu, X. Zhou, H. Tong, and M. Mahdavi, "Do we really need complicated model architectures for temporal networks?" 2 2023. [Online]. Available: http://arxiv.org/abs/2302.11636

[16] P. Atkins, J. de Paula, and J. Keeler, *Atkins' Physical Chemistry*, 11th ed. Oxford University Press, 2018.

[17] L. Pauling, *The Nature of the Chemical Bond, An Introduction to Modern Structural Chemistry*, 3rd ed. Cornell University Press, 1960.

[18] K. J. Knight, "Pharma chemistry," *Pharmaceutical Journal*, vol. 282, pp. 105–128, 1 2021.

[19] R. Taylor, O. Kennard, and W. Versichel, "The geometry of the $n - h|o = c$ hydrogen bond. 3. hydrogen-bond distances and angles," *urn:issn:0108-7681*, vol. 40, pp. 280–288, 6 1984. [Online]. Available: //scripts.iucr.org/cgi-bin/paper?a23216

[20] M. H. Anowar, A. Shamail, A. J. Albertsen, B. Hall, T. J. Thielen, B. Riemersma, and G. Trajcevski, "Detecting and visualizing bond-forming convoys in atomic and molecular trajectories," in *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 657–660. [Online]. Available: https://doi.org/10.1145/3678717.3691264

[21] D. E. McRee, "Computational techniques," *Practical Protein Crystallography*, pp. 91–cp1, 1 1999.

[22] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ArXiv*, vol. abs/2010.11929, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:225039882

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[24] S. Narang, H. W. Chung, Y. Tay, L. Fedus, T. Fevry, M. Matena, K. Malkan, N. Fiedel, N. Shazeer, Z. Lan, Y. Zhou, W. Li, N. Ding, J. Marcus, A. Roberts, and C. Raffel, "Do transformer modifications transfer across implementations and applications?" in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 5758–5773. [Online]. Available: https://aclanthology.org/2021.emnlp-main.465/

[25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[26] X. Wang, D. W. House, P. A. Oroskar, A. Oroskar, A. Oroskar, C. J. Jameson, and S. Murad, "Md simulations of the chiral recognition mechanism for a polysaccharide chiral stationary phase in enantiomeric chromatographic separations," *An International Journal at the Interface Between Chemistry and Physics*, vol. 117, 12 2019.

[27] X. Wang, C. J. Jameson, and S. Murad, "Modeling enantiomeric separations as an interfacial process using amylose tris(3,5-dimethylphenyl carbamate) (admpc) polymers coated on amorphous silica," *Langmuir*, vol. 36, pp. 1113–1124, 2 2020.

[28] A. Guerrero-Corella, A. Fraile, and J. Alemán, "Intramolecular hb activation: Strategies, benefits, and influence in catalysis," *ACS Organic & Inorganic Au*, 2022.

[29] D. D. Bibelayi, A. S. Lundemba, P. V. Tsalu, P. I. Kilunga, J. M. Tshishimbi, and Z. G. Yav, "Hydrogen bonds of c=s, c=se and c=te with c-h in small-organic molecule compounds derived from the cambridge structural database (csd)," 2021.

[30] E. S. Wibowo and B.-D. Park, "Two-dimensional nuclear magnetic resonance analysis of hydrogen-bond formation in thermosetting crystalline urea–formaldehyde resins at a low molar ratio," *ACS Applied Polymer Materials*, 1 2022.

[31] Y. Tian, Y. Qi, and F. Guo, "Freedyg: Frequency enhanced continuous-time dynamic graph model for link prediction," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=82Mc5ilInM

[32] K. Huang, C. Xiao, L. M. Glass, M. Zitnik, and J. Sun, "Skipgnn: predicting molecular interactions with skip-graph networks," *Scientific reports*, vol. 10, no. 1, p. 21092, 2020.

[33] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 1025–1035.

[34] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *International Conference on Learning Representations*, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:5959482

[35] H. Mao, J. Li, H. Shomer, B. Li, W. Fan, Y. Ma, T. Zhao, N. Shah, and J. Tang, "Revisiting link prediction: A data perspective," 2024. [Online]. Available: https://arxiv.org/abs/2310.00793

[36] T. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *ArXiv*, vol. abs/1609.02907, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:3144218

[37] M. Zhang, P. Li, Y. Xia, K. Wang, and L. Jin, “Labeling trick: a theory of using graph neural networks for multi-node representation learning,” in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, ser. NIPS ’21. Red Hook, NY, USA: Curran Associates Inc., 2021.

[38] M. Zhang and Y. Chen, “Link prediction based on graph neural networks,” in *Advances in Neural Information Processing Systems*, 2018, pp. 5165–5175.

[39] Z. Zhu, Z. Zhang, L.-P. Xhonneux, and J. Tang, “Neural bellman-ford networks: a general graph neural network framework for link prediction,” in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, ser. NIPS ’21. Red Hook, NY, USA: Curran Associates Inc., 2021.

[40] H. Shomer, Y. Ma, H. Mao, J. Li, B. Wu, and J. Tang, “Lpformer: An adaptive graph transformer for link prediction,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 8 2024, pp. 2686–2698.

[41] Y. Luo, X. Zhao, J. Zhou, J. Yang, Y. Zhang, W. Kuang, J. Peng, L. Chen, and J. Zeng, “A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information,” *Nature Communications*, vol. 8, no. 1, sep 2017. [Online]. Available: https://doi.org/10.1038/s41467-017-00680-8

[42] J. Y. Ryu, H. U. Kim, and S. Y. Lee, “Deep learning improves prediction of drug–drug and drug–food interactions,” *Proceedings of the National Academy of Sciences*, vol. 115, no. 18, pp. E4304–E4311, 2018. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1803294115

[43] M. Zitnik, M. Agrawal, and J. Leskovec, “Modeling polypharmacy side effects with graph convolutional networks,” *Bioinformatics*, vol. 34, no. 13, pp. i457–i466, 06 2018. [Online]. Available: https://doi.org/10.1093/bioinformatics/bty294

[44] F. Wan, L. Hong, A. Xiao, T. Jiang, and J. Zeng, “Neodti: neural integration of neighbor information from a heterogeneous network for discovering new drug–target interactions,” *Bioinformatics*, vol. 35, no. 1, pp. 104–111, 07 2018. [Online]. Available: https://doi.org/10.1093/bioinformatics/bty543

[45] Z. Zheng, J. Qi, J. Li, G. Chao, J. Dong, and Y. Yu, “Efficient discovery of motif transition process for large-scale temporal graphs,” 4 2025. [Online]. Available: http://arxiv.org/abs/2504.15979

[46] V. S. Dave and M. Al Hasan, “Triangle completion time prediction using time-conserving embedding,” in *Machine Learning and Knowledge Discovery in Databases*, U. Brefeld, E. Fromont, A. Hotho, A. Knobbe, M. Maathuis, and C. Robardet, Eds. Cham: Springer International Publishing, 2020, pp. 541–557.