Chemically-Informed Geometric Graph Alignment for Molecular Captioning with Large Language Models

Enzo Azzoug
enzo.azzoug@eleves.enpc.fr

Bronislav Abadie
bronislav.abadie@eleves.enpc.fr

Pablo Larrieu
pablo.larrieu@eleves.enpc.fr

January 15, 2026

Abstract

Molecular graph captioning aims to translate the complex topological and geometric information of molecules into coherent natural language descriptions. In this work, we propose a novel multimodal architecture that integrates explicit 3D geometric priors and chemical knowledge into a Graph Neural Network (GNN) encoder, aligned with a Large Language Model (Qwen 2.5) via Low-Rank Adaptation (LoRA). We introduce a chemically-driven node sorting mechanism to deterministically break permutation invariance, creating positional semantics for the attention heads. Furthermore, we implement a rigorous geometric feature extraction pipeline capturing spatial conformations via spherical coordinates. Our approach optimizes a dual objective: a contrastive alignment loss for latent space structure and a causal modeling loss for text generation. Crucially, we treat atoms as a sequence of visual tokens projected via a Multi-Layer Perceptron (MLP), preserving fine-grained structural resolution for the LLM.

1 Introduction

Representing molecules as graphs is standard in chemoinformatics, yet standard Message Passing Neural Networks (MPNNs) often fail to capture the subtle 3D geometric nuances—such as chirality, cis/trans isomerism, and steric hindrance—that determine chemical function [1]. While recent advances in Large Language Models (LLMs) offer powerful generative capabilities, bridging the gap between the continuous latent space of geometric graphs and the discrete token space of LLMs remains an open challenge [2].

Our contribution is threefold:

1. A **geometric preprocessing pipeline** that encodes spherical coordinates and bond priorities to enrich edge features, explicitly handling 3D conformers.

2. A **hierarchical encoder architecture** utilizing Pre-Norm Multi-Head Attention and a concatenation of Laplacian (LPE), Random Walk (RWSE), and Weisfeiler-Lehman (WL) positional encodings [3].

3. An efficient **multimodal training strategy** combining contrastive pre-training (Graph-Text matching) and LoRA-based generative fine-tuning [4], avoiding global pooling to preserve atomic resolution.

2 Methodology

2.1 Chemically-Informed Geometric Preprocessing

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a molecular graph where $v_i \in \mathcal{V}$ represents an atom. The dataset provides raw 3D spatial coordinates $\mathbf{p}_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ for each atom.

2.1.1 Raw Feature Embeddings

Before any geometric processing, we transform the discrete chemical attributes into continuous vectors. For each atom v_i, we have a set of categorical attributes $\mathcal{A}_i = \{a_i^{(z)}, a_i^{(c)}, a_i^{(d)}, \dots\}$ representing the atomic number, chirality tag, degree, formal charge, etc. We use learnable embedding matrices $\mathbf{E}^{(k)}$ for each attribute type k. The initial raw atom feature $\mathbf{x}_{\text{atom},i}$ is obtained by summing these embeddings:

$$\mathbf{x}_{\text{atom},i} = \sum_{k \in \text{Attributes}} \mathbf{E}^{(k)}[a_i^{(k)}] \in \mathbb{R}^{d_{emb}} \tag{1}$$

The matrix $\mathbf{X}_{\text{atom}} \in \mathbb{R}^{N \times d_{emb}}$ collects these vectors for all atoms in the molecule. Similarly, for each edge e_{ij}, categorical attributes (bond type, stereochemistry) are embedded to form $\mathbf{e}_{ij}^{\text{cat}}$, which serves as a basis for edge-conditioned operations.

2.1.2 Deterministic Neighbor Sorting via Chemical Priors

To define a consistent receptive field for the Transformer, we transform the unordered neighborhood set $\mathcal{N}(i)$ into a **canonical ordered sequence**. For a central node i, we define a priority tuple S_{ij} for each neighbor j:

$$S_{ij} = (\mathcal{T}(e_{ij}),\ Z_j,\ |C_j|,\ R_j) \tag{2}$$

where $\mathcal{T}(e_{ij})$ is the bond priority (Triple > Double > ...), Z_j is the atomic number, $|C_j|$ is the absolute formal charge, and $R_j \in \{0, 1\}$ indicates if the atom is part of a ring.

The sorting process is lexicographical and strictly deterministic. For any two neighbors $u, v \in \mathcal{N}(i)$, the ordering is decided as follows:

1. First, we compare the bond priorities: if $\mathcal{T}(e_{iu}) > \mathcal{T}(e_{iv})$, then u precedes v.
2. If there is a tie, we compare the atomic numbers: if $Z_u > Z_v$, then u precedes v.
3. Then, we compare the absolute formal charges: if $|C_u| > |C_v|$, then u precedes v.
4. Finally, we prioritize atoms within rings: if $R_u > R_v$, then u precedes v.

This ensures that the "first neighbor" seen by the attention head always corresponds to the most chemically significant connection (e.g., a double bond to an Oxygen), effectively breaking the permutation invariance of standard GNNs. We simulate a position encoding that usually works well with transformers architectures.

2.1.3 Spherical Geometric Features

We compute a geometric feature vector $\mathbf{g}_{ij} \in \mathbb{R}^5$ for each edge using relative vectors $\mathbf{r}_{ij}$:

$$\mathbf{g}_{ij} = [\|\mathbf{r}_{ij}\|, \sin(\phi_{ij}), \cos(\phi_{ij}), \sin(\theta_{ij}), \cos(\theta_{ij})]^\top \tag{3}$$

2.2 Hierarchical Encoder Architecture

2.2.1 Augmented Node Initialization

We initialize the node feature matrix $\mathbf{H}^{(0)} \in \mathbb{R}^{N \times d}$ by projecting the raw atom embeddings and fusing them with structural encodings. We define the operation in vectorized form:

$$\mathbf{H}^{(0)} = \mathbf{X}_{\text{atom}}\mathbf{W}_{\text{atom}} + [\mathbf{E}_{\text{RW}} \oplus \mathbf{E}_{\text{LPE}}]\mathbf{W}_{\text{glob}} + \mathbf{E}_{\text{WL}}\mathbf{W}_{\text{wl}} \tag{4}$$

where $\mathbf{X}_{\text{atom}}$ is the categorical embedding matrix defined in Sec. 2.1.1, $\oplus$ is the concatenation along the feature dimension, and $\mathbf{W}$ are learnable linear projections. This fuses local chemistry, subgraph topology (RWSE/WL), and global geometry (LPE).

2.2.2 Geometric Multi-Head Attention

The encoder consists of L layers using **Pre-Norm Multi-Head Attention (MHA)**. Let H be the number of attention heads. For a given layer l, head h, and central node i, let $\mathbf{N}_i \in \mathbb{R}^{M \times d}$ be the matrix of features of its sorted neighbors.

The attention logic incorporates both the geometric features $\mathbf{g}_{ij}$ and the categorical edge embeddings $\mathbf{e}_{ij}^{\text{cat}}$. We define a unified structural bias vector $\mathbf{b}_{ij} = \text{Concat}(\mathbf{g}_{ij}, \mathbf{e}_{ij}^{\text{cat}})$.

$$\mathbf{q}_i^{(h)} = \mathbf{x}_{\text{norm},i}\mathbf{W}_Q^{(h)} \quad \in \mathbb{R}^{d_h} \tag{5}$$

$$\mathbf{K}_{\mathcal{N}(i)}^{(h)} = \mathbf{N}_i\mathbf{W}_K^{(h)} \quad \in \mathbb{R}^{M \times d_h} \tag{6}$$

$$\mathbf{V}_{\mathcal{N}(i)}^{(h)} = \mathbf{N}_i\mathbf{W}_V^{(h)} \quad \in \mathbb{R}^{M \times d_h} \tag{7}$$

The attention scores are biased by a projection of the structural features:

$$\mathbf{e}_i^{(h)} = \frac{\mathbf{q}_i^{(h)}(\mathbf{K}_{\mathcal{N}(i)}^{(h)})^\top}{\sqrt{d_h}} + (\mathbf{B}_i\mathbf{u}^{(h)})^\top \in \mathbb{R}^M \tag{8}$$

where $\mathbf{B}_i$ stacks the structural bias vectors $\mathbf{b}_{ij}$ for the sorted neighbors, and $\mathbf{u}^{(h)}$ is a learnable projection vector for head h.

The final update aggregates all heads and projects back:

$$\mathbf{h}_i^{(l)} = \mathbf{h}_i^{(l-1)} + \left[\mathbf{o}_i^{(1)} \oplus \cdots \oplus \mathbf{o}_i^{(H)}\right] \mathbf{W}_O \tag{9}$$

2.3 LLM Integration via LoRA

We use **Qwen2.5-1.5B-Instruct** as the decoder [5].

2.3.1 Visual Token Sequence Projection

We preserve atomic resolution by avoiding global pooling. Let $\mathbf{H}^{(L)} \in \mathbb{R}^{N \times d}$ be the output of the graph encoder. We project this sequence to the LLM dimension d_{llm} via an MLP:

$$\mathbf{Z}_G = \text{Linear}_2\left(\sigma_{\text{GELU}}\left(\text{Linear}_1(\mathbf{H}^{(L)})\right)\right) \in \mathbb{R}^{N \times d_{llm}} \tag{10}$$

The full input sequence for the LLM is the concatenation $\mathbf{E}_{\text{input}} = [\mathbf{Z}_G, \mathbf{E}_{\text{text}}]$.

2.3.2 LoRA Formulation

Due to the significant memory footprint of the 1.5B parameter model, full fine-tuning is computationally intractable on our available hardware. Therefore, we freeze the pre-trained weights $\mathbf{W}_0 \in \mathbb{R}^{d_{out} \times d_{in}}$ in the LLM and inject trainable rank-decomposition matrices:

$$\mathbf{W} = \mathbf{W}_0 + \frac{\alpha}{r}\mathbf{B}\mathbf{A} \tag{11}$$

with $\mathbf{B} \in \mathbb{R}^{d_{out} \times r}$, $\mathbf{A} \in \mathbb{R}^{r \times d_{in}}$, $r = 8$ and $\alpha = 16$. This strategy allows us to adapt the linguistic capabilities of the LLM to the specific syntax of chemical descriptions without catastrophic forgetting or memory overflow.

3 Training Objectives

The training optimizes two losses sequentially to ensure robust modality alignment before text generation.

3.1 Phase 1: Contrastive Alignment

We align the mean-pooled graph representation $\bar{\mathbf{u}}$ with the text representation $\bar{\mathbf{v}}$ using InfoNCE. For a batch of size B, with similarity matrix $\mathbf{S} = \mathbf{U}\mathbf{V}^\top/\tau$:

$$\mathcal{L}_{\text{NCE}} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\mathbf{S}_{ii})}{\sum_{j=1}^{B} \exp(\mathbf{S}_{ij})} \tag{12}$$

Justification: This contrastive loss is critical for structuring the multimodal latent space. By pulling the embeddings of matching molecule-description pairs together and pushing non-matches apart, we ensure that the encoder learns semantically meaningful representations before the LLM begins the generation task. This mitigates the "cold start" problem where the projector would otherwise map noise to the LLM during the first steps of optimization.

3.2 Phase 2: Generative Modeling

We minimize the negative log-likelihood over the token sequence $y = (y_1, \ldots, y_T)$:

$$\mathcal{L}_{\text{Gen}} = -\sum_{t=1}^{T} \log P_\theta(y_t \mid y_{<t}, \mathbf{Z}_G) \tag{13}$$

Justification: We employ the Causal Language Modeling (CLM) loss to leverage the autoregressive nature of the pre-trained LLM. This forces the model to learn the conditional probability distribution of the chemical description given the sequence of visual tokens $\mathbf{Z}_G$, effectively translating the geometric insights of the encoder into natural language.

4 Experimental Setup

4.1 Hyperparameters

For the training of the auto-encoder described before we used the following hyperparameters 1. We didn't spend much time to fine tune them and we also had a constraint with the memory.

Parameter	Value
Graph Embedding Dim (d)	256
Attention Heads (H)	4
Neighbors (M)	11
LLM Embedding Dim	1536
Batch Size	32
Projector Hidden Dim	1536
LoRA Rank (r)	8
LoRA Alpha (α)	16

Table 1: Hyperparameters used for training.

4.2 Evaluation of the encoder

Table 2 presents the quantitative performance of our encoder to find the good embedding in the latent space neighborhood. The goal was for each graph of the validation set to verify if we find the label (embedding) when we look for the K nearest neighbors from the set in the latent space. We observe that for $K = 1$ we get an accuracy of 27.5% and for $K = 10$ we get 60.7% which means that the latent space is quite well structured.

K	**1**	**5**	**10**
Accuracy (%)	27.5	51.4	60.7

Table 2: Encoder Performances on R@K

4.3 Evaluation of the decoder

Model	BLEU-4	ROUGE-L	Average length
Ours	0.1331	0.2903	36.73
Baseline	0.1483	0.3001	41.58

Table 3: Model Performance on Validation Set

Table 3 shows our model evaluated on the validation set compared to the baseline. The latter slightly outperforms our model on BLEU-4 and ROUGE-L, indicating higher n-gram overlap with the reference descriptions, likely due to its retrieval-based nature. Our model generates somewhat shorter descriptions on average, which may reduce surface-level overlap despite producing fluent and chemically plausible text.

5 Limitations

Our results are constrained by hardware limitations (VRAM), which restricted batch sizes and training steps. Qualitatively, although captions are chemically coherent, we observe "semantic drift" where valid synonyms penalize n-gram metrics like BLEU. Additionally, our deterministic sorting is heuristic-based; learning the optimal permutation remains an open challenge.

6 Conclusion

We presented a framework aligning 3D geometric graphs with LLMs using spherical coordinates and deterministic node sorting. By projecting atomic sequences directly into Qwen 2.5 via LoRA, we preserve structural fidelity. Future work will focus on scaling the training and exploring learnable sorting mechanisms.

References

[1] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural Message Passing for Quantum Chemistry," *International Conference on Machine Learning (ICML)*, 2017.

[2] C. Edwards, T. Lai, K. Ros, G. Honke, and H. Ji, "Translation between Molecules and Natural Language," *arXiv preprint arXiv:2204.11817*, 2022.

[3] V. P. Dwivedi and X. Bresson, "A Generalization of Transformer Networks to Graphs," *arXiv preprint arXiv:2012.09699*, 2020.

[4] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," *International Conference on Learning Representations (ICLR)*, 2022.

[5] J. Bai et al., "Qwen Technical Report," *arXiv preprint arXiv:2309.16609*, 2023.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *Advances in Neural Information Processing Systems (NIPS)*, 2017.