Supplementary Information of Towards practical molecular structure elucidation: A lightweight multi-spectral CNN-Transformer framework

Qinghai Cui*, Álvaro Cimas, Marie-Pierre Gaigeot

August 17, 2026

Contents

1 Introduction of Model Methodology

1.1 Vocabulary

The vocabulary of QM9s IR, Raman and QM9-NMR is {'PAD': 0, 'BOS': 1, 'EOS': 2, 'C': 3, 'N': 4, 'O': 5, 'F': 6, '1': 7, '2': 8, '3': 9, '4': 10, '5': 11, '=': 12, '#': 13, '(': 14, ')': 15}.

The vocabulary of NIST IR is {'PAD': 0, 'BOS': 1, 'EOS': 2, 'C': 3, 'N': 4, 'O': 5, 'F': 6, 'Br': 7, 'Cl': 8, 'I': 9, 'P': 10, 'S': 11, '1': 12, '2': 13, '3': 14, '4': 15, '5': 16, '=': 17, '#': 18, '(': 19, ')': 20}.

The vocabulary of Open-source Raman is {'PAD': 0, 'BOS': 1, 'EOS': 2, 'C': 3, 'N': 4, '1': 5, '(': 6, ')': 7, '=': 8, 'O': 9, 'Br': 10, 'Cl': 11, 'S': 12, '#': 13}.

1.2 Loss Function

The loss function of base model is cross entropy loss. Cross entropy loss is widely adopted in classification tasks to quantify the discrepancy between the predicted probability distribution and the ground-truth label distribution. Let $y = [y_1, y_2, \ldots, y_C]$ denote the ground-truth vector, where C is the total number of classes. The model outputs class-wise predicted probabilities $\hat{y} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_C]$ after the Softmax activation:

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log \hat{y}_i \tag{1}$$

satisfying $\sum_{i=1}^{C} \hat{y}_i = 1$ and $0 < \hat{y}_i < 1$. For the fine-tuning part, unlikelihood training is used to resolve specific failure modes where the model makes error like miscounting the multiplicity of structural motifs. Given the ground-truth sequence of tokens $\mathbf{y} = (y_1, y_2, \ldots, y_T)$ and the input spectrum $\mathbf{X}$, the model is trained to maximize the log-likelihood of the correct next token given the preceding context:

$$L_{\text{CE}} = -\sum_{t=1}^{T} \log P(y_t \mid y_{<t}, \mathbf{X}; \theta) \tag{2}$$

where θ represents the model parameters.

To prevent penalizing the correctly generated shared prefix, the unlikelihood penalty is localized exclusively to the divergent tokens. The localized Unlikelihood Loss minimizes the probability assigned to the incorrect tokens from step d onward:

$$L_{\text{UL}} = \frac{1}{|\mathbf{M}|} \sum_{t=d}^{|\hat{\mathbf{y}}|} -\log\left(1 - p_t(\hat{y}_t)\right) \tag{3}$$

where $p_t(\hat{y}_t) = P(\hat{y}_t \mid \hat{\mathbf{y}}_{<t}, \mathbf{X}; \theta)$ is the predicted probability of the erroneous token, and $|\mathcal{M}|$ is the number of unmasked tokens from the divergence point to the sequence end. To ensure numerical stability, $(1 - p_t)$ is clamped to a lower

bound of 10^{-8}. The final training objective is a weighted summation of both losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{UL}}\mathcal{L}_{\text{UL}} \tag{4}$$

1.3 Exact Match

The final output sequences are generated token by token. The output accuracy is done with exact match, where it truncates both predicted token sequences and ground-truth label sequences at the first EOS (end-of-sequence) token before conducting element-wise full-sequence matching. Only sequences identical up to and including the EOS token are counted as correct predictions. Greedy search and beam search methods are also common choices for benchmark results comparison.

1.4 Invalid Output and Duplications

Even if the prediction sequence is invalid, it is still considered as a result even in beam search top-k ranks. Also the duplicated sequence outputs are not deleted. The duplications indicate that the model is quite confident at such output. The evaluation part has no RDKit included for re-ranking to make sure that our results are shown without any adjustment.

1.5 Standard Positional Encoding

In the model, positional encoding is used with a standard PyTorch implementation of the sine/cosine positional encoding proposed in the original Transformer paper:

$$x_{\text{out}} = \text{Dropout}\big(x_{\text{embed}} + \mathbf{PE}\big) \tag{5}$$

It behaves as a fixed buffer without learnable parameters. The dropout layer is employed to reduce overfitting and stabilizing training procedure.

1.6 SE Channel Attention

The SE channel attention is 1D convolutional squeeze-and-excitation attention with fused global average pooling and max pooling.

$$\begin{aligned}
\mathbf{s}_{\text{avg}} &= \text{AdaptiveAvgPool1d}(\mathbf{x}) \in R^{B\times C\times 1}, \quad \tilde{\mathbf{s}}_{\text{avg}} = \text{squeeze}(\mathbf{s}_{\text{avg}}) \\
\mathbf{s}_{\text{max}} &= \text{AdaptiveMaxPool1d}(\mathbf{x}) \in R^{B\times C\times 1}, \quad \tilde{\mathbf{s}}_{\text{max}} = \text{squeeze}(\mathbf{s}_{\text{max}}) \\
\mathbf{z} &= \text{MLP}(\tilde{\mathbf{s}}_{\text{avg}}) + \text{MLP}(\tilde{\mathbf{s}}_{\text{max}}) \\
\alpha &= \mathit{Sigmoid}(\mathbf{z}) \in R^{B\times C} \\
\mathbf{x}_{\text{out}} &= \mathbf{x} \odot \text{unsqueeze}(\alpha, -1)
\end{aligned} \tag{6}$$

where B is batch, C is channel, $\odot$ denotes element-wise Hadamard product. An example of Hadamard production is:

$$\begin{bmatrix} 2 & 3 & 1 \\ 0 & 8 & -2 \end{bmatrix} \odot \begin{bmatrix} 3 & 1 & 4 \\ 7 & 9 & 5 \end{bmatrix} = \begin{bmatrix} 2\times 3 & 3\times 1 & 1\times 4 \\ 0\times 7 & 8\times 9 & -2\times 5 \end{bmatrix} = \begin{bmatrix} 6 & 3 & 4 \\ 0 & 72 & -10 \end{bmatrix}.$$

1.7 Transformer with Gated Attention Residual

Unlike standard residual connections, the gated attention scales the attention contribution by a learnable scalar parameter γ, which is passed through a hyperbolic tangent activation:

$$\begin{aligned} g &= \tanh(\gamma), \quad \gamma = \text{block\_gate} \in \mathbf{R}^1 \\ \mathbf{x}_{\text{out}} &= \mathbf{x}_{\text{identity}} + g \cdot \mathbf{h}_{\text{trans}} \end{aligned} \tag{7}$$

where:

- γ: Single learnable scalar parameter of the transformer block, initialized to $\gamma = 0$;
- g: Gate openness coefficient mapped from γ via hyperbolic tangent activation;
- $\mathbf{x}_{\text{identity}}$: Original input feature tensor of the block;
- $\mathbf{h}_{\text{trans}}$: Feature tensor processed by positional encoding, multi-head self-attention and feed-forward network.

Gate openness refers to the scaling factor g that controls the contribution of transformer-processed feature $\mathbf{h}_{\text{trans}}$ to final output. We use three representative values of learnable parameter $\gamma \in \{-1, 0, 1\}$ to demonstrate distinct openness states:

1. $\gamma = -1$ $g = \tanh(-1) \approx -0.7616$, **Negative Moderate Openness** The transformer branch feature is subtracted from original input; the model suppresses attention-extracted sequential patterns.
2. $\gamma = 0$ (initialization state) $g = \tanh(0) = 0$, **Fully Closed Gate (Zero Openness)** $g \cdot \mathbf{h}_{\text{trans}} = \mathbf{0}$, output exactly equals raw input. At training start, block acts as identity mapping to stabilize gradient.
3. $\gamma = 1$ $g = \tanh(1) \approx 0.7616$, **Positive Moderate Openness** Transformer feature positively enhances original input; sequential attention information dominates representation.

Boundary limit cases:

- $\gamma \to +\infty \Rightarrow g \to 1$: Fully positive open gate, maximum transformer feature weight.
- $\gamma \to -\infty \Rightarrow g \to -1$: Fully reverse open gate, maximum transformer feature suppression.

1.8 Attention Mechanism

In our model architecture, it leverages multi-head attention. For a given set of queries ($\mathbf{Q}$), keys ($\mathbf{K}$), and values ($\mathbf{V}$), the scaled dot-product attention for a single head i is defined as:

$$\text{head}_i = \text{softmax}\left(\frac{(\mathbf{Q}\mathbf{W}_i^Q)(\mathbf{K}\mathbf{W}_i^K)^T}{\sqrt{d_k}}\right)\mathbf{V}\mathbf{W}_i^V \tag{8}$$

where $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V$ are learnable parameter matrices, and d_k is the dimensionality of the keys. The outputs of all h heads are concatenated and subjected to a final linear projection:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}^O \tag{9}$$

where $\mathbf{W}^O$ represents the output weight matrix.

The patched attention is used at the start of the model. It splits the continuous single-channel raw signal into non-overlapping patch tokens via 1D convolution patch embedding to largely reduce computational cost:

$$\begin{aligned} \mathbf{h}_{\text{patch}} &= \text{Conv1d}_{\text{patch}}(\mathbf{x}) \\ \mathbf{h} &= \mathbf{h}_{\text{patch}}.\text{transpose}(1,2) \end{aligned} \tag{10}$$

where $\mathbf{h} \in \mathbf{R}^{B \times N_p \times D_p}$, $N_p = \dfrac{L}{\text{patch\_size}}$, $D_p = \text{patch\_dim}$. All self-attention and feed-forward operations run only on patch-level tokens instead of original full-length time steps. And after transformer computation, transposed convolution unpatchifies feature back to original signal length.

Cross attention between encoder-decoder part is:

$$\text{head}_i^{\text{cross}} = \text{softmax}\left(\frac{(\mathbf{Q}_{\text{dec}}\mathbf{W}_i^Q)(\mathbf{K}_{\text{enc}}\mathbf{W}_i^K)^T}{\sqrt{d_k}}\right)\mathbf{V}_{\text{enc}}\mathbf{W}_i^V \tag{11}$$

where $\mathbf{K}_{\text{enc}}$ and $\mathbf{V}_{\text{enc}}$ represent encoder memory comes from the encoder, while $\mathbf{Q}_{\text{dec}}$ is decoder hidden states coming from the decoder.

1.9 Settings of Parameters

$t0\_head = 1$
$t1\_head = 2$
$t2\_head = 4$
$t3\_head = 8$
$epochs = 200$
$d\_model = 256$
$en\_layers = 6$
$de\_layers = 6$
$en\_head = 8$

$de\_head = 8$
$en\_dim_ffeed = 1024$
$de\_dim_ffeed = 1024$
$dropout = 0.1$
$max\_len = 100$
$patience = 10$ # based on valid loss, best model .pt file is selected with best valid accuracy
$batch\_size = 32/16$ # uni-spectroscopy 32, joint spectra 16

2 Introduction of Detailed Analysis Methodology

2.1 Causal Ablation

In this work, we tested the multimodal input of concatenated IR, Raman, and NMR spectra. Zero ablation test is done to quantitatively disentangle the independent contribution of each spectroscopy to final prediction performance of output sequence. Let the concatenated input tensor be

$$\mathbf{X} = \left[\mathbf{X}_{\text{IR}},\ \mathbf{X}_{\text{Raman}},\ \mathbf{X}_{\text{NMR}}\right].$$

When ablating IR, it equals

$$\mathbf{X}_{\text{zero,IR}} = \left[\mathbf{0},\ \mathbf{X}_{\text{Raman}},\ \mathbf{X}_{\text{NMR}}\right].$$

The same replacement rule applies for Raman-only ablation:

$$\mathbf{X}_{\text{zero,Raman}} = \left[\mathbf{X}_{\text{IR}},\ \mathbf{0},\ \mathbf{X}_{\text{NMR}}\right]$$

and NMR-only ablation is

$$\mathbf{X}_{\text{zero,NMR}} = \left[\mathbf{X}_{\text{IR}},\ \mathbf{X}_{\text{Raman}},\ \mathbf{0}\right].$$

We quantify the predictive contribution of each modality by computing its share of overall performance drop, denoted as modality attribution share in percentage. Let D_{full} represent the baseline accuracy obtained from complete ternary input $\mathbf{X} = \left[\mathbf{X}_{\text{IR}}, \mathbf{X}_{\text{Raman}}, \mathbf{X}_{\text{NMR}}\right]$. For zero ablation of modality m, the degraded accuracy is written as $D_{\text{zero},m}$. The total aggregated performance drop across all three modalities is defined as:

$$\Delta_{\text{total}} = \left(D_{\text{full}} - D_{\text{zero,IR}}\right) + \left(D_{\text{full}} - D_{\text{zero,Raman}}\right) + \left(D_{\text{full}} - D_{\text{zero,NMR}}\right). \tag{12}$$

The percentage performance share attributed to modality m is calculated by normalizing its individual accuracy drop against the total aggregated drop:

$$\text{Share}_m(\%) = \frac{D_{\text{full}} - D_{\text{zero},m}}{\Delta_{\text{total}}} \times 100. \tag{13}$$

The formula above holds for each spectroscopic input: IR, Raman and NMR, which yields three percentage values summing to exactly 100%. A larger $\text{Share}_m(\%)$ indicates the corresponding modality carries a larger proportion of discriminative chemical information for model prediction.

2.2 Attention Rollout

The attention rollout is used for analysis as the global overview of encoder attention mechanism.

$$\mathbf{A}_{i,j}^{(l)} = \text{softmax}\left(\frac{(\mathbf{Q}\mathbf{W}^Q)(\mathbf{K}\mathbf{W}^K)^T}{\sqrt{D_p}}\right)_{i,j} \tag{14}$$

$$\mathbf{R}^{(1)} = \mathbf{A}^{(1)}$$

Input token correlations are first captured by $\mathbf{A^{(1)}}$.

$$\mathbf{R}^{(l)} = \mathbf{A}^{(l)} \cdot \mathbf{R}^{(l-1)} \tag{15}$$

$$\mathbf{R}^{(6)} = \mathbf{A}^{(6)} \cdot \mathbf{A}^{(5)} \cdot \mathbf{A}^{(4)} \cdot \mathbf{A}^{(3)} \cdot \mathbf{A}^{(2)} \cdot \mathbf{A}^{(1)} \tag{16}$$

The sequential matrix multiplication $\mathbf{A}^{(6)} \cdot \mathbf{A}^{(5)} \cdot \mathbf{A}^{(4)} \cdot \mathbf{A}^{(3)} \cdot \mathbf{A}^{(2)} \cdot \mathbf{A}^{(1)}$ physically represents signal transmission through the entire encoder stack.

$$w_k = \mathbf{R}_{t,k}^{(6)} \tag{17}$$

where t denotes the index of output prediction readout token, k is input patch token index. After computing token-level cumulative attribution weights from the fully propagated 6-layer corrected attention rollout matrix, we aggregate all patch tokens according to their source spectroscopic modality to derive modality-wise contribution percentages, which enable direct quantitative comparison with the causal ablation modality attribution derived from zero/mean ablation experiments.

$$W_m = \sum_{k \in \text{modality } m} w_k \tag{18}$$

$$W_{\text{total}} = W_{\text{IR}} + W_{\text{Raman}} + W_{\text{NMR}} \tag{19}$$

$$\text{RolloutShare}_m(\%) = \frac{W_m}{W_{\text{total}}} \times 100 \tag{20}$$

2.3 Saliency

As a dedicated learnable starting token, the <BOS>token acts as a global aggregation anchor: the model compresses all discriminative spectral features into this single token for downstream regression/classification. <BOS>-based saliency quantifies the direct pairwise dependency weight between the special input and the final output scalar Y^{BOS}.

$$S_{BOS}(X) = \left| \frac{\partial Y^{BOS}}{\partial X} \right| \tag{21}$$

After computing <BOS>-based saliency map $S_{BOS}(\mathbf{X})$, we aggregate saliency values by modality to obtain comparable percentage attribution share matching ablation and attention rollout metrics. Let S_m represent the summed gradient saliency magnitude of all feature entries belonging to modality m (IR / Raman / NMR):

$$S_m = \sum_{k \in \text{modality } m} S_{BOS}(\mathbf{X})_k \tag{22}$$

Total aggregated saliency magnitude across all three spectroscopic inputs:

$$S_{\text{total}} = S_{\text{IR}} + S_{\text{Raman}} + S_{\text{NMR}} \tag{23}$$

Modality attribution percentage from BOS gradient saliency:

$$\text{GradSaliencyShare}_m(\%) = \frac{S_m}{S_{\text{total}}} \times 100 \tag{24}$$

2.4 Sequence Saliency

Instead of only evaluate based on the <BOS>token, sequence saliency is performed for evaluation of per-spectroscopy importance based on whole output sequence. Sequence saliency is defined as element-wise absolute partial derivative of prediction against token sequence embeddings:

$$S_c(\mathbf{T}) = \left| \frac{\partial Y^c}{\partial \mathbf{T}} \right| \tag{25}$$

Let S_k stand for the summed gradient magnitude of all embedding dimensions of patch token k:

$$S_k = \sum_{dim} S_c(\mathbf{T})_{k,dim} \tag{26}$$

Aggregate token-level sequence saliency by modality group (IR / Raman / NMR):

$$S_m = \sum_{k \in \text{modality } m} S_k \tag{27}$$

The final share attribution results are calculated as same as the the <BOS>-only saliency.

2.5 GradCAM

Although both focus on the starting token <BOS>only, Gradient-weighted Class Activation Mapping (GradCAM) targets the final output feature tensor of the last 1D convolution layer inside the CNN stem part, distinct from saliency. This method restricts gradient computation exclusively to the CNN layer, gradients do not propagate to raw IR/Raman/NMR input, nor do we extract gradients from Transformer encoder layers.

The GradCAM calculation is done with computing channel-averaged gradient weights first, where gradients are derived exclusively from the <BOS>-based prediction scalar Y^{BOS}

$$\alpha_d = \frac{1}{N_p} \sum_{t=1}^{N_p} \frac{\partial Y^{BOS}}{\partial \mathbf{F}_{d,t}} \tag{28}$$

α_d = importance weight for CNN feature F channel d, averaged over all patch spatial positions t. All partial derivatives are computed against the prediction signal sourced solely from the <BOS>starting token. Weighted summation of CNN feature channels to generate raw GradCAM activation map driven by <BOS>prediction

$$M_{\text{GradCAM}} = \sum_{d=1}^{D_p} \alpha_d \cdot \mathbf{F}_{d,:} \tag{29}$$

Then negative activations are rectified to retain only positive predictive contribution to the <BOS>token

$$M^{+}_{\text{GradCAM}} = \max\left(M_{\text{GradCAM}},\ \mathbf{0}\right) \tag{30}$$

After aggregating GradCAM activation magnitude by source modality (IR / Raman / NMR), the contribution to the ¡BOS¿ aggregation is measured:

$$M_t = M^{+}_{\text{GradCAM}}(t) \tag{31}$$

$$M_m = \sum_{t \in \text{modality } m} M_t \tag{32}$$

where M_t represent summed positive GradCAM activation for patch spatial position t.

$$M_{\text{total}} = M_{\text{IR}} + M_{\text{Raman}} + M_{\text{NMR}} \tag{33}$$

$$\text{GradCAMShare}_m(\%) = \frac{M_m}{M_{\text{total}}} \times 100 \tag{34}$$

2.6 Gradient Activation

With all gradient signals exclusively originated from the starting <BOS>, the spectroscopy attribution of layer-wise gradient-weighted activation share is calculated by extracting values from every convolutional layer in the CNN stem part.

$$\alpha_d^{(l\text{cnn})} = \frac{1}{N_{l\text{cnn}}} \sum_{t=1}^{N_{l\text{cnn}}} \frac{\partial Y^{BOS}}{\partial \mathbf{F}_{d,t}^{(l\text{cnn})}} \tag{35}$$

$\alpha_d^{(l\text{cnn})}$ = channel importance coefficient for channel d of CNN layer l_{cnn} with feature F, averaged over all spatial patch positions t. Every partial derivative links back to the <BOS>starting point. The followed steps are as same as the GradCAM method.

$$M_{\text{raw}}^{(l\text{cnn})} = \sum_{d=1}^{D_{l\text{cnn}}} \alpha_d^{(l\text{cnn})} \cdot \mathbf{F}_{d,:}^{(l\text{cnn})} \tag{36}$$

$$M_{+}^{(l\text{cnn})} = \max\left(M_{\text{raw}}^{(l\text{cnn})},\ \mathbf{0}\right) \tag{37}$$

$$M_t^{(l\text{cnn})} = M_{+,t}^{(l\text{cnn})} \tag{38}$$

$$M_m^{(l\text{cnn})} = \sum_{t \in \text{modality } m} M_t^{(l\text{cnn})} \tag{39}$$

$$M_{\text{total}}^{(l\text{cnn})} = M_{\text{IR}}^{(l\text{cnn})} + M_{\text{Raman}}^{(l\text{cnn})} + M_{\text{NMR}}^{(l\text{cnn})} \tag{40}$$

$$\text{CNNLayerShare}_m^{(l\text{cnn})}(\%) = \frac{M_m^{(l\text{cnn})}}{M_{\text{total}}^{(l\text{cnn})}} \times 100 \tag{41}$$

2.7 Integrated Gradient

The integrated gradient calculates different global gradients from the targeted <BOS>output with varied interpolated inputs. The interpolated input is uniformly scaled from zero baseline to full signal. Here we used 10 steps interpolation. $\mathbf{X}_{\text{baseline}} = \mathbf{0}$: zero-value baseline tensor.

$$n_{\text{step}} = 10 \tag{42}$$

∇_α = global input gradient, computed against the raw network input $\mathbf{X}_\alpha$.

$$\mathbf{X}_\alpha = \frac{\alpha}{10}\mathbf{X}, \quad \alpha = 1, \ldots, 10 \tag{43}$$

$$\nabla_\alpha = \frac{\partial Y_{\text{BOS-only}}^c}{\partial \mathbf{X}_\alpha} \tag{44}$$

$$\text{IG}(\mathbf{X}) = \mathbf{X} \cdot \frac{1}{10} \sum_{\alpha=1}^{10} \nabla_\alpha \tag{45}$$

$$Share_{\text{IG}}(\mathbf{X}) = |\text{IG}(\mathbf{X})| \tag{46}$$

This method is similar to saliency method. However, saliency uses only one single gradient sample and lacks core theoretical guarantees, while integrated gradient integrates gradient information across a full zero-to-full-signal interpolation path to fix saturation artifacts and satisfy complete attribution axioms.

2.8 Decoder Attention Evolution

We also performed a per-layer corss-attention multi-spectral attribution analysis for every independent decoder layer targeted on the starting token <BOS>.

$$\mathbf{A}^{(l)} = \text{CrossAttnWeight}^{(l)} \tag{47}$$

where $\mathbf{A}^{(l)} \in \mathbf{R}^{B\times H\times 1\times T\text{enc}}$, raw multi-head cross-attention matrix for decoder layer l. B = batch size, H = number of decoder attention heads, T_{enc} = compressed encoder token length.

$$\overline{\mathbf{A}}^{(l)} = \frac{1}{H}\sum_{h=1}^{H} \mathbf{A}^{(l)}_{:,h,BOS,:} \tag{48}$$

$\overline{\mathbf{A}}^{(l)} \in \mathbf{R}^{B\times T\text{enc}}$ is head-averaged attention for the first <BOS>target point.

$$\mathbf{I}^{(l)} = \text{Interpolate}\big(\overline{\mathbf{A}}^{(l)},\ L_{\text{total}}\big) \tag{49}$$

Then the attention is aligned with original full spectral input through $\mathbf{I}^{(l)} \in \mathbf{R}^{B\times L}\text{total}$.

$$A_m^{(l)} = \sum_{pos\in\text{modality } m} \mathbf{I}^{(l)}_{pos} \tag{50}$$

$A_m^{(l)}$ = total cross-attention attribution magnitude of modality m in decoder layer l.

$$A^{(l)}_{\text{total}} = A^{(l)}_{\text{IR}} + A^{(l)}_{\text{Raman}} + A^{(l)}_{\text{NMR}} \tag{51}$$

Modality attribution percentage from BOS attention for layer l is:

$$\text{DecAttnShare}_m^{(l)}(\%) = \frac{A_m^{(l)}}{A^{(l)}_{\text{total}}} \times 100 \tag{52}$$

3 Parameter Change Comparison between Base and Fine-tuned Models

To understand the structural impact of UL fine-tuning on the model, we analyzed the parameter-space drift across all distinct architectural regions. We quantified this change using two complementary metrics. First, to understand how much each layer was altered relative to its original state, we calculated the scale-free relative drift ($||\Delta\theta||/||\theta_{base}||$). Second, to determine which components absorbed the bulk of the actual fine-tuning updates, we calculated each region's share of the total update budget, represented by the squared norm of the parameter change ($||\Delta\theta||^2$). The results are shown in Figure S18. As shown in subgraph (a), when adjusting for the initial scale of the weights, both the Encoder and Decoder layers exhibit the highest relative drift (ranging from

0.080 to 0.090). Decoder L0 and the early Encoder layers (L0-L2) experienced the most significant proportional shifts. Conversely, the CNN components, normalization layers, and target embeddings remained remarkably stable, showing minimal relative deviation from their base states. However, the subgraph (b) —— the total share of the update budget ($||\Delta\theta||^2$), tells a different story: The later layers of the Decoder (L1 through L5) overwhelmingly dominated the fine-tuning process, collectively absorbing nearly 45% of the total update magnitude. Decoder L3 alone accounted for 9.4% of the total change. These results indicate that while the early Encoder layers underwent significant relative changes, their overall contribution to the absolute update budget was moderate (roughly 6% to 7% per layer). The fine-tuning process primarily dedicated its parameter-space updates to reshaping the deeper layers of the Decoder, while leaving foundational feature extractors (like the CNN stem part) practically untouched.

4 <BOS>Token Analysis

Both graphs display the exact same two-dimensional Principal Component Analysis (PCA) projection of <BOS>for the molecular dataset. The data points form a few distinct clusters: a large, curved distribution on the left side (negative PC1), a dispersed cluster in the top right, and denser, smaller clusters in the bottom right. The difference between the two graphs is purely how these identical data points are color-coded to reveal different chemical properties.

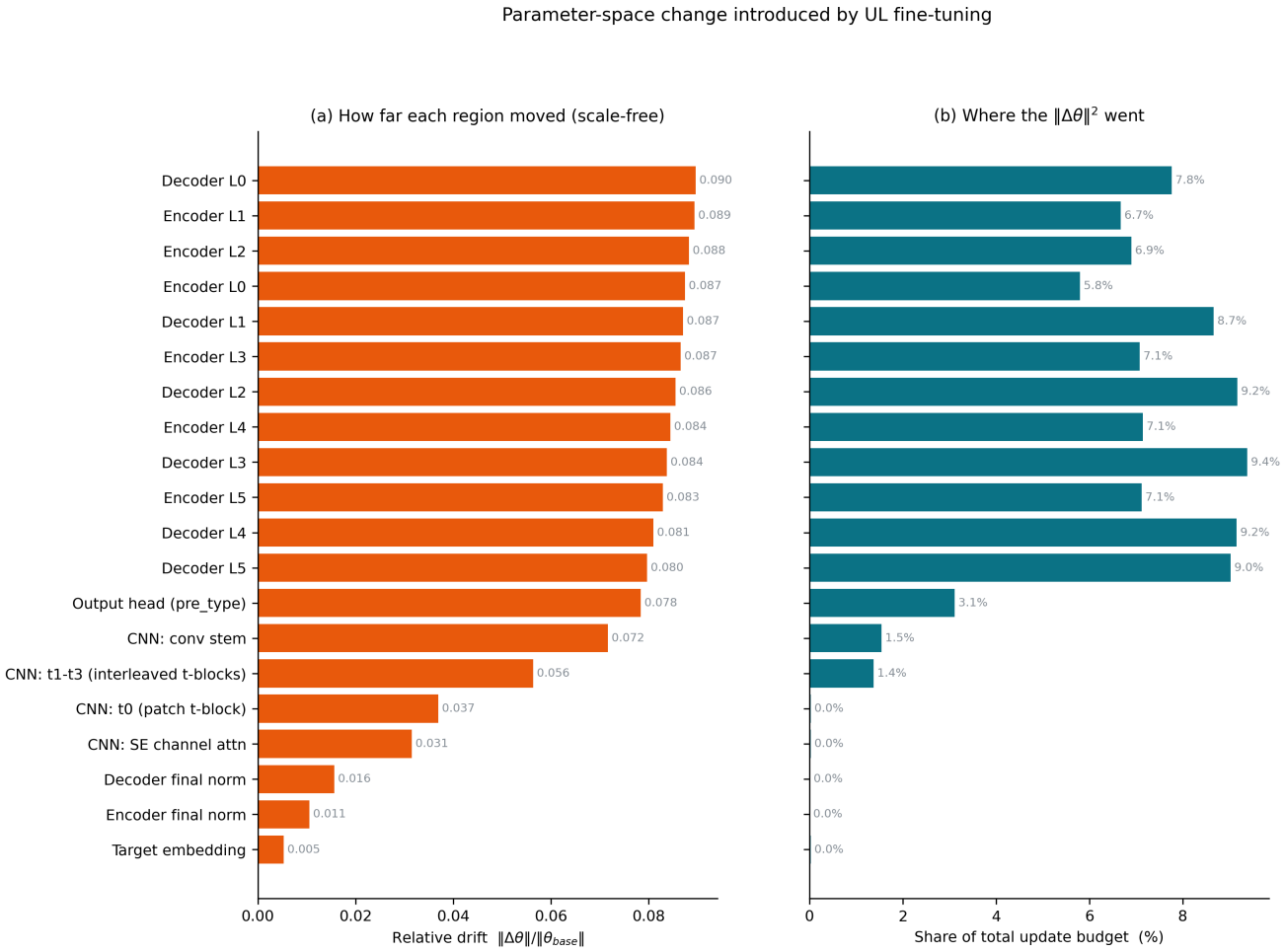

Figure S1: Parameter drifts of different regions with total share of the parameter update budget

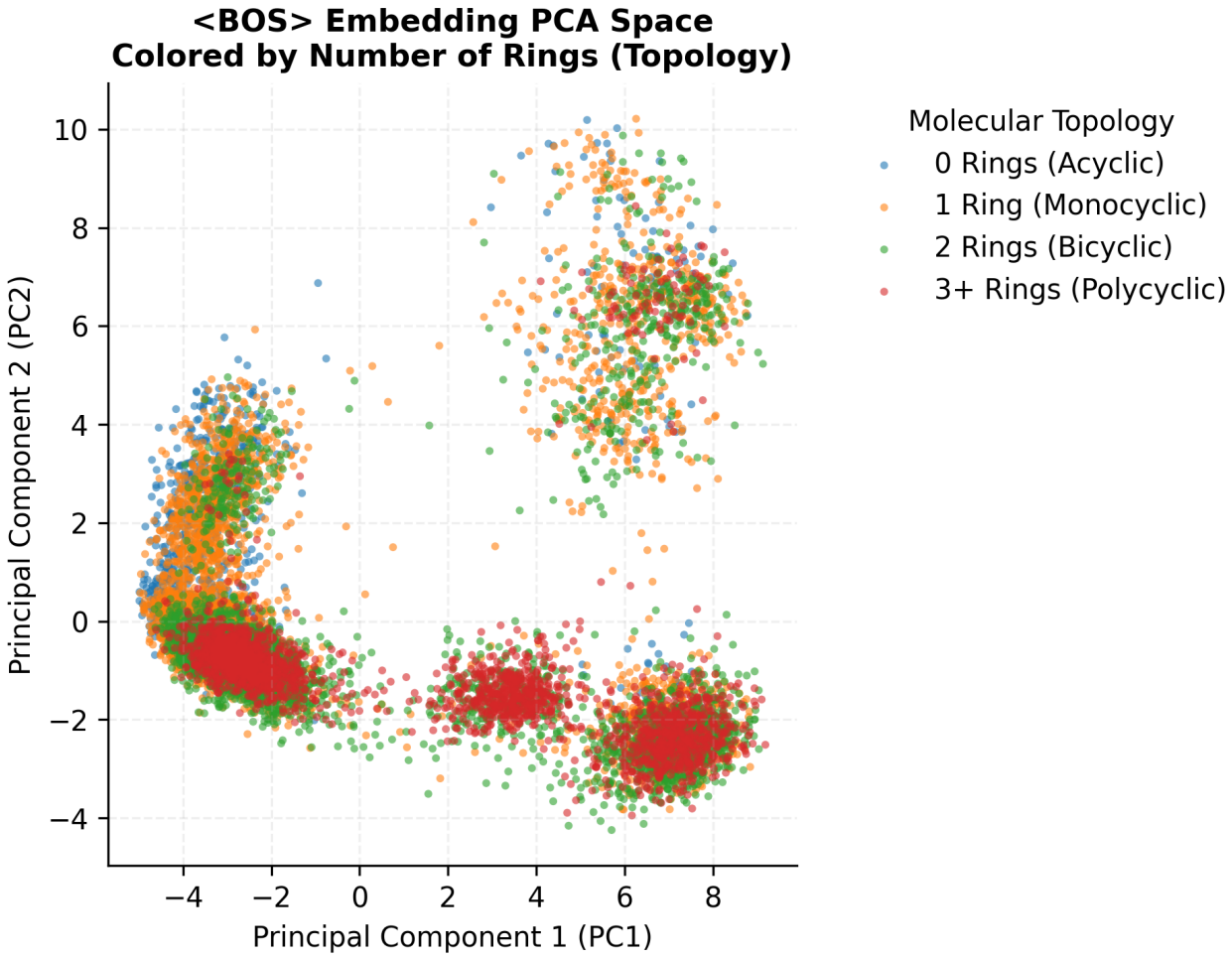

Figure S2: PCA analysis of molecular ring differences onto the <BOS>token scalar output

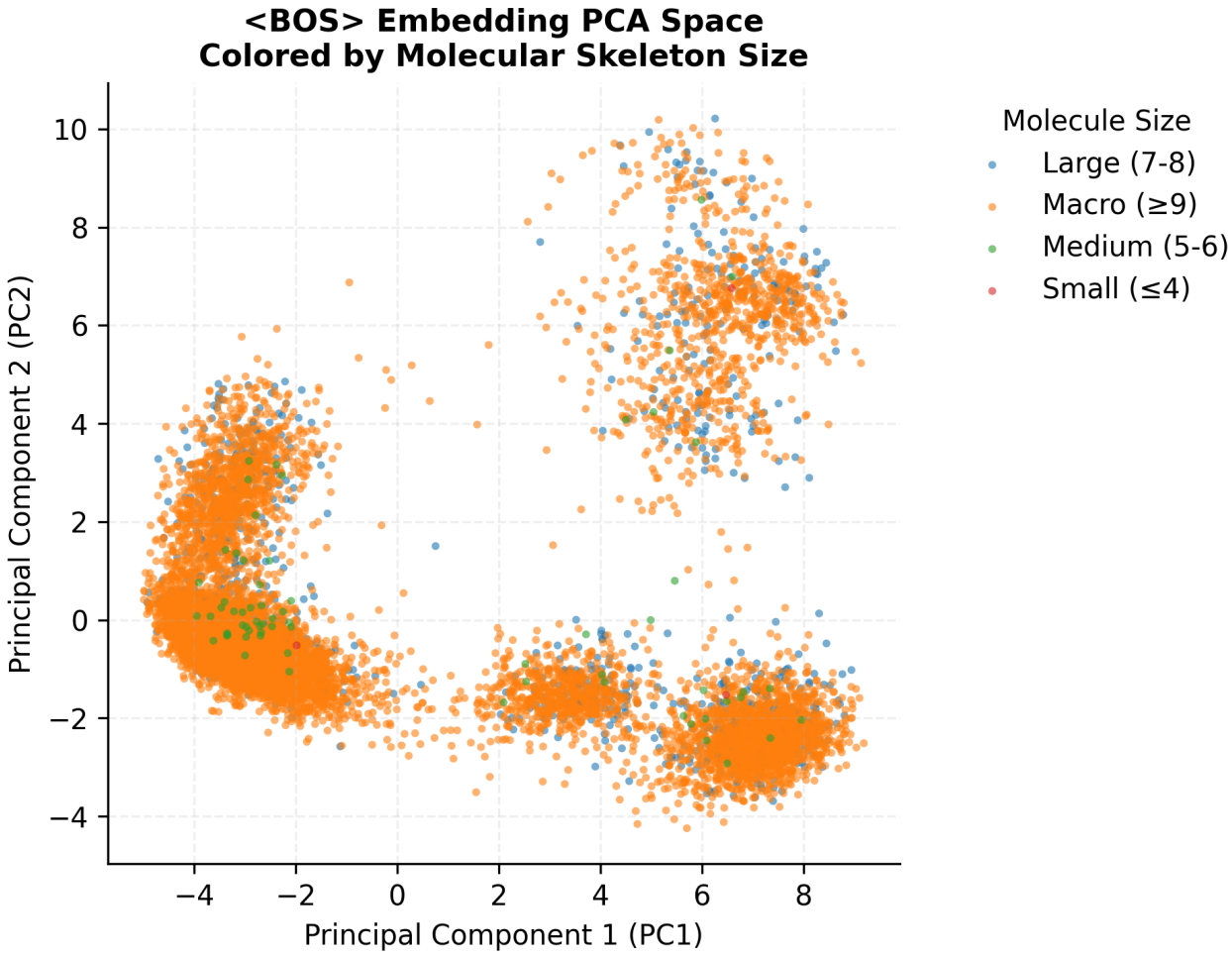

Figure S3: PCA analysis of molecular size differences onto the <BOS>token scalar output