DISTANCE IS MORE THAN A BIAS: GEOMETRY-DRIVEN VALUE MIXING IN GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Geometric deep learning and graph transformers have driven significant advancements in learning from structured data, particularly in molecular property prediction and spatial modeling. However, while recent approaches often incorporate spatial distances as attention biases or complex pair representations, this significantly increases computational overhead. In this paper, we propose a paradigm shift: rather than treating distance merely as a bias, we utilize the distance matrix as a direct value-mixing operator to achieve both high predictive performance and extreme computational efficiency. We introduce DisAttention, which replaces standard query-key attention with mixing weights derived from learnable transformations of the distance matrix. To further enhance expressive power, we propose DisMix, which integrates node feature affinity through symmetric normalization and distance gating instead of conventional row-wise softmax. By eliminating the need to update high-dimensional pair states, our geometry-driven approach drastically simplifies computation. Extensive experiments across molecular datasets (QM9, MolPCBA, LP-PDBbind) and broader geometric domains (MNIST superpixels, ModelNet40) demonstrate that our models achieve highly competitive accuracy while significantly reducing parameter count, peak memory usage, and training time compared to state-of-the-art graph transformers.

1 INTRODUCTION

Geometric deep learning and graph transformers have achieved important results on a range of problems in chemistry and biology, including molecular property prediction (Gilmer et al., 2017; Fang et al., 2022) and biomolecular interaction modeling. Graph transformers have achieved strong performance by applying modern deep learning architectures to graph problems, encoding relations between nodes in attention weights to pass information between node representations. To combine their strengths, geometric deep learning with graph transformers has incorporated spatial distance information into attention and node updates in various forms, including continuous filters, attention biases, and pair representations (Maziarka et al., 2020; Thölke & De Fabritiis, 2022). Although their designs differ, prior studies share the goal of transforming given spatial relations into mixing operators to construct node or graph representations useful for prediction. This leads to the question of whether a given distance matrix can be transformed directly into a mixing operator to achieve both predictive performance and computational efficiency, rather than expanding distance information into more complex relational representations.

We propose a method that goes beyond using distance as a bias in attention scores by constructing the operator that mixes node values directly from a distance matrix. Our **DisAttention** model replaces the attention matrix with mixing weights obtained through a learnable transformation of the distance matrix, without using queries or keys. With DisAttention, we empirically show that an appropriately transformed distance matrix itself can form a mixing operator and achieve predictive performance competitive with graph transformers that use content-based attention.

Furthermore, **DisMix** adds node feature affinity based on conventional query-key attention to DisAttention, forming a more expressive mixing operator. Unlike row-softmax-based attention, DisMix does not apply row-wise softmax to attention scores. Instead, it applies symmetric normalization to an affinity that combines distance and content, then adjusts the mixing weights with a distance gate. Because no further normalization is applied after gating, distance information influences both

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

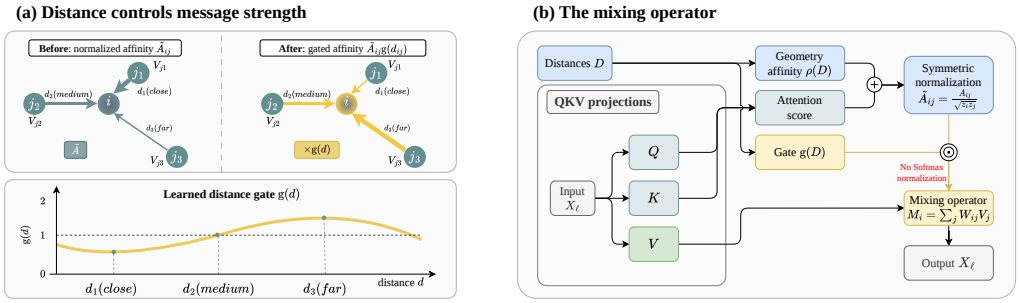


Figure 1: Distance-driven value mixing. (a) Distance gating adjusts normalized mixing weights and message strength. (b) DisMix combines geometry and content before symmetric normalization and gating, then aggregates values.

the relative mixing proportions between nodes and the overall magnitude of the information passed. DisMix achieved competitive performance on the MolPCBA dataset with substantially fewer parameters than existing SOTA models.

The proposed distance-driven mixing simplifies the computations needed to construct and maintain representations of relations between nodes, offering substantial computational cost advantages while retaining performance competitive with existing models. Both models introduce a distance-matrix-driven mixing operator directly into the attention process, reducing the burden of having to represent node-node interactions through updates to high-dimensional pair representations across layers. We compare the efficiency of these designs with graph transformers of similar parameter counts on QM9 and LP-PDBbind. We evaluate practical computational efficiency through parameter counts, training step time, and peak memory, and analyze actual training progress by comparing performance improvements over both wall-clock time and epochs.

We also combine DisMix with a pretrained graph transformer to examine the applicability of our method to existing architectures and pretrained models, and extend evaluation to MNIST superpixel and ModelNet40 to empirically confirm its versatility across architectures and domains. Over the first 20 epochs on QM9 gap, the validation errors of both models decreased more steeply than that of Graphormer, with log–log decay rates of approximately 0.29 versus 0.043. Our models achieved accuracies of up to $98.90 \pm 0.09\%$ on MNIST superpixel and 83.47% on ModelNet40, confirming the effectiveness of the same mixing structure for non-molecular geometries. Finally, we use ablations and visualizations to analyze how the learned radial bias, symmetric normalization, and distance gate affect the behavior and predictive performance of the mixing operator.

Our main contributions are as follows.

- We propose DisAttention, which constructs mixing weights from a distance matrix without queries or keys, and demonstrate its competitive advantage over graph transformers with content-based attention under limited compute.
- We propose DisMix as a mixing method that reintroduces node feature affinity while firmly imposing geometric weighting through symmetric normalization and a distance gate, and demonstrate both strong performance and computational efficiency.
- We evaluate predictive performance, computational cost, and convergence speed across diverse graphs and geometries, and validate the scope of the proposed approach and the roles of its components through integration with a pretrained backbone and analysis of the operators.

2 Related Works

Continuous-filter and geometric message passing. Incorporating spatial distances into learnable message functions has been widely studied in geometric deep learning. SchNet generates continuous filters from interatomic distances to compute channel-wise weighted sums of node features (Schütt et al., 2017). EGNN computes messages from node features and squared distances

and combines them with relative coordinate updates to preserve $E(n)$-equivariance (Satorras et al., 2021). DimeNet uses distances and angles to update directional message embeddings (Gasteiger et al., 2020). The DisGNN study analyzes the expressivity limitations of node-level message passing based on pairwise distances and uses k-tuple-based k-DisGNN to exploit higher-order geometry encoded in distance matrices (Li et al., 2023). These studies show that geometry-dependent message passing can be learned without query–key attention. We connect this distance-based message modulation to the mixing structure of graph transformers, constructing an operator that acts directly on node values through head-specific scalar weights, normalization, and gating.

Distance and Graph Transformers. Graphormer incorporates graph structure into a transformer through centrality encoding based on node degree and attention biases that encode shortest-path distances and edge information (Ying et al., 2021). Using queries Q, keys K, and values V obtained from node representations, mixing in this bias-based attention can be expressed as follows (Vaswani et al., 2017; Ying et al., 2021).

$$\alpha = \mathrm{softmax}_{\mathrm{row}}\left(\frac{QK^{\top}}{\sqrt{d_h}} + B\right), \qquad M = \alpha V, \tag{1}$$

Here, α is the attention matrix containing mixing weights between node values, and M is the message passed through it. d_h is the head dimension, and B is a bias that adjusts query–key scores by incorporating graph structure, such as distances and edge information.

EGT constructs attention biases and gates by projecting edge representations, updated at each layer, into head-specific scalars (Hussain et al., 2022). TGT updates pair representations containing distance and edge information through triplet interactions and incorporates them into the biases and gates of node attention (Hussain et al., 2024). GPTrans generates layer-specific attention biases from updated edge representations and propagates node and edge information jointly through node-to-edge and edge-to-node propagation (Chen et al., 2023). Uni-Mol+ maintains atom and pair representations, updates pairs with triangular operations, and iteratively refines coordinates (Lu et al., 2024). These designs use structural information flexibly but require computations to maintain and update high-dimensional pair states in addition to node representations. In molecular prediction, they are also sometimes combined with domain-specific input processing, including bond features, distance encodings based on atom-pair types, and RDKit-based geometry. Uni-Mol uses 3D molecular pretraining with distance-based pair representations (Zhou et al., 2023), while Transformer-M learns from both 2D and 3D molecular data (Luo et al., 2023). We focus on constructing a mixing operator directly from a distance matrix without updating high-dimensional pair states. DisAttention removes query–key comparisons (cf. Tay et al., 2021), while DisMix reintroduces content affinity but applies a distance-only gate after symmetric normalization. This allows geometry to contribute directly to affinity construction and message strength adjustment.

3 Distance-based Mixing

In equation 1, the attention matrix α acts as a mixing operator between node values. We replace it with an operator W constructed from a distance matrix and extend this approach by incorporating node content. Let N be the number of nodes and d the hidden width, with node representations denoted by $X \in \mathbb{R}^{N\times d}$. For node i with coordinates r_i, the input distance matrix is $D_{ij} = \|r_i - r_j\|_2$.

3.1 DisAttention: Geometry-Driven Value Mixing

Rather than estimating relations between nodes through query–key comparisons, DisAttention learns weights for mixing values according to the given distances. Values are obtained in the same way as in conventional transformers. The mixing weights W are constructed using two distance functions instead of query–key comparisons.

$$\rho(t) = \log G(t) + b(t), \qquad g(t) = 2\sigma(q(t)). \tag{2}$$

Here, t is the pairwise distance, and σ is the sigmoid function. G is a Gaussian with a learnable center and width for each head, learning distance ranges useful for interactions. The learned radial bias b adjusts the distance response of the Gaussian, while q generates the sigmoid input for the distance gate g. b and q are computed by separate MLPs that take the same 64 RBF basis functions

as input, with distinct parameters for each layer and head-specific outputs. Thus, the two functions use the same distance information but learn different roles: forming the normalized affinity and subsequently adjusting message strength.

We define the mixing operator from these distance functions as follows.

$$\begin{aligned} A_{ij} &= \exp(\rho(D_{ij}))\,, \qquad z_i = \sum_j A_{ij}, \\ \widetilde{A}_{ij} &= \frac{A_{ij}}{\sqrt{z_i z_j}}, \qquad\qquad W_{ij} = \widetilde{A}_{ij}\, g(D_{ij}). \end{aligned} \tag{3}$$

Here, A is a positive affinity obtained from distances, and $\widetilde{A}$ is the result of symmetric normalization using the node affinity sums z_i (Kipf & Welling, 2017). The final message is obtained as $M = WV$, or $M_i = \sum_j W_{ij} V_j$. This allows W to learn not only individual distances but also the distribution of distances within a graph to weight values, while, unlike attention, being determined entirely by geometry, independently of node features. This construction thus retains the transformer's value propagation and node updates while removing the Q/K pathway that computes mixing weights between nodes.

3.2 DisMix: Distance Affinity with Content Bias

DisAttention determines relations through geometry and passes node features as values, but it could be more expressive if it could adjust the degree of mixing according to node types and their current representations, even for the same spatial arrangement. To this end, DisMix introduces query–key scores as a bias in the distance-based log-affinity. That is, instead of using distance to adjust content-based attention, it adjusts an affinity constructed from distances according to content.

For head dimension d_h, the query–key score S and its symmetrized content score C are defined as

$$S_{ij} = \frac{Q_i^\top K_j}{\sqrt{d_h}}, \qquad C_{ij} = \begin{cases} (S_{ij} + S_{ji})/2, & i \neq j, \\ 0, & i = j \end{cases} \tag{4}$$

The query–key scores are symmetrized to preserve the symmetry of distance-based relations after adding content. Diagonal entries are removed only from the content term, while the distance affinity at $D_{ii} = 0$ is retained. With a head-specific learnable coefficient λ controlling the influence of the content score on the affinity, the affinity of DisMix is given by

$$A_{ij} = \exp(\rho(D_{ij}) + \lambda C_{ij}) = \exp(\rho(D_{ij}))\exp(\lambda C_{ij})\,, \qquad \lambda = \tanh\beta \tag{5}$$

where β is a head-specific learnable parameter. This equation distinguishes the roles of the distance-based affinity and the content-dependent multiplicative adjustment.

The subsequent symmetric normalization and distance gate are the same as in equation 3. When content changes A_{ij}, it also changes z_i, z_j used for normalization, adjusting not only the relation within that pair but also those between the two nodes and their other neighbors. In contrast, the final gate $g(D_{ij})$ remains a function of distance alone, adjusting distance-specific message strength on top of the content-adaptive affinity. DisMix computes these adjustments from node representations updated at each layer and does not pass a separate high-dimensional pair representation to the next layer, yielding a less complex architecture and cost advantages over existing models.

Normalization and Message Strength. Both models apply a distance gate after symmetric normalization without renormalization. Symmetric normalization adjusts weights using the affinity sums of both nodes while preserving symmetry, but does not fix each row sum to 1 as softmax does in standard attention. The distance gate can therefore change both the relative mixing proportions and the message scale for each node. To see this, fix the affinity and values and consider the case where all gates have the same constant value c, $0 < c < 2$:

$$\frac{cA_{ij}}{\sqrt{(cz_i)(cz_j)}} = \widetilde{A}_{ij}, \qquad M^{(g\equiv c)} = c\, M^{(g\equiv 1)} \tag{6}$$

The left-hand side corresponds to incorporating the gate into the affinity before normalization, whereas the right-hand side represents the message obtained by applying the gate after normalization. The former cancels the common scale c, while the latter directly adjusts the message magnitude while preserving the relative mixing proportions. Row-normalizing the final W also removes

this common scale. By applying the gate without renormalization, DisAttention and DisMix learn distance-specific transmission strengths on top of the normalized affinity.

4 EXPERIMENTS

We present several stages of experiments to evaluate how simple and compact distance-driven mixing can make the model architecture while retaining competitive predictive performance. First, we compare predictive performance, model size, and actual execution costs on MolPCBA, achieving SOTA-compatible performance with relatively few parameters. We demonstrate the cost efficiency and versatility of our models through their training convergence speed and extension to vision datasets. Finally, we use qualitative analysis to examine how distance-driven mixing learns meaningful information from the given data.

4.1 PREDICTIVE PERFORMANCE AND COMPUTATIONAL COST

We compare test average precision (AP) and parameter counts on MolPCBA (Hu et al., 2020) to evaluate whether distance-driven mixing can reduce model size while retaining strong predictive performance. Table 1 compares representative graph transformers with two sizes of DisMix.

Model	Test AP (%) ↑	Parameters (M)	AP/M ↑
EGT-Larger[1,a]	29.61±0.24	110.8	0.267
Graphormer-FLAG[2]	31.39±0.32	119.5	0.263
TGT-Ag + RDKit[3]	31.44±0.29	47.0	0.669
GRPE-Large[1,b]	31.50±0.10	118.3	0.266
TGT-Ag + TGT-At-DP[3]	31.67±0.31	47.0	0.674
GraphGPT-B24[1]	*31.81±0.10*	227.3	0.140
GPTrans-L[4]	**32.43±0.22**	86.0	0.377
DisAttention (3D, seed 1)	23.33	5.6	**4.189**
DisMix (13M)	30.44±0.29	13.0	*2.338*
DisMix + GPTrans-L	30.98±0.12	82.8	0.374

Table 1: Predictive performance and model size on MolPCBA.
Bold: best, *italics*: second best (in each performance column). AP/M is AP (%) divided by the parameter count in millions.
Result sources: [1]Zhao et al. (2025). [2]Ying et al. (2021). [3]Hussain et al. (2024). [4]Chen et al. (2023).
Model sources: [a]Hussain et al. (2022); [b]Park et al. (2022).

DisMix achieved a test AP of $30.44 \pm 0.29\%$ across three seeds. Although this is 1.00, 1.37, and 0.95 %p below TGT-Ag + RDKit, GraphGPT-B24, and Graphormer-FLAG, respectively, its parameter count is substantially smaller, at approximately 28%, 6%, and 11% of theirs, respectively. DisMix has an AP/M of 2.338, compared with 0.674 for TGT-Ag + TGT-At-DP, a ratio of approximately 3.47. Considering model size alongside the differences in absolute AP, this shows that strong predictive performance can be achieved with a small parameter budget. The large DisMix model with a shared pretrained node-processing module transplanted from GPTrans-L has 82.784M parameters and achieves an AP of $30.98 \pm 0.12\%$. The configuration with approximately 6.36 times as many parameters provides an additional gain of 0.54 percentage points, revealing both the potential to use large representations and the performance–size trade-off.

4.1.1 RUNTIME AND MEMORY

To determine whether a reduction in parameter count translates into lower actual execution costs, Table 2 and Figure 2 compare training step time and peak training memory measured on an RTX 5090.

On a common batch of four 26-atom molecules, DisMix has a training step time of 37.20ms and peak memory of 0.308GiB, the second-lowest among the configurations in the table. Its step time and memory are 22.2% and 69.4% lower than those of TGT-Ag + RDKit, and 56.1% and 88.6% lower than those of Graphormer-FLAG, respectively. Thus, for this input, parameter savings translate into lower memory usage during actual training, with a larger reduction in memory than in step time. This

Model	Training step (ms) ↓	Peak memory (GiB) ↓
1. EGT-Larger	70.58	2.497
2. Graphormer-FLAG	84.83	2.712
3. TGT-Ag + RDKit	47.80	1.005
4. GRPE-Large	49.05	2.663
5. TGT-Ag + DP	43.78	1.005
6. GraphGPT-B24	62.06	5.219
7. GPTrans-L	45.66	1.950
8. DisAttention	**19.18**	**0.141**
9. DisMix	*37.20*	*0.308*
10. DisMix + GPTrans-L	72.05	1.870

Table 2: Training step time and peak allocated memory measured for the models on the same 26-atom batch.

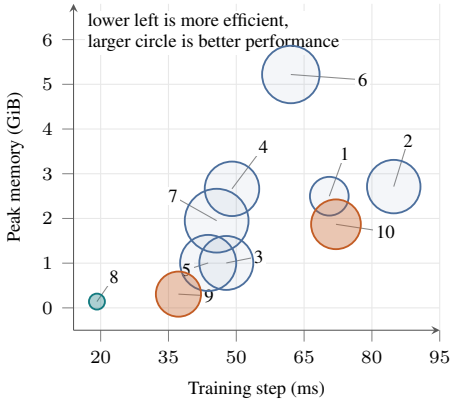


Figure 2: Positions toward the lower left indicate lower step and memory costs, and larger circles indicate higher AP in Table 1.

is consistent with the design, in which dense value mixing and node update computations remain even as the model becomes smaller. DisMix, located in the lower left of Figure 2, offers an option that reduces both step time and memory cost in exchange for somewhat lower absolute AP.

4.2 Learning Cost Efficiency

Even if the computational cost per step is small, the advantage in total training cost diminishes if more training is required to reach a target level of performance. We therefore train on QM9 (Ramakrishnan et al., 2014) and LP-PDBbind (Li et al., 2026) for up to 180 epochs and compare predictive error, error reduction across epochs, and training wall-clock time together.

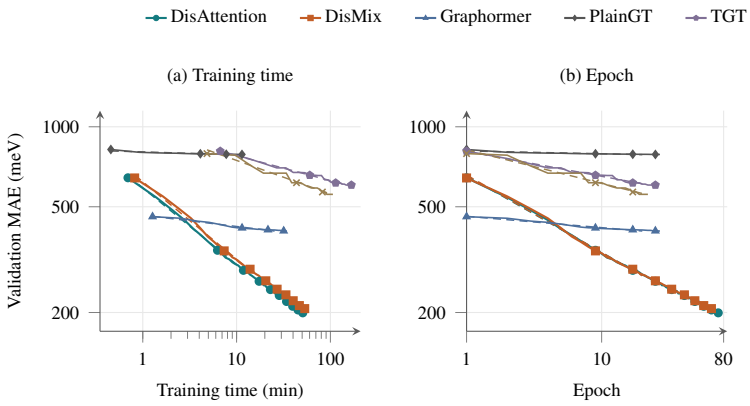
	QM9: Test MAE ↓						LP-PDBbind
Model	μ D	α bohr^3	HOMO eV	LUMO eV	Gap meV	$\langle R^2 \rangle$ bohr^2	RMSE ↓ pK
PlainGT	0.9125±0.0006	3.413±0.004	0.3873±0.0007	0.7401±0.0019	793.43±0.58	178.04±0.02	1.6611±0.0430
Graphormer	0.7263±0.0122	1.345±0.043	0.2364±0.0026	0.3285±0.0071	399.04±2.17	112.30±1.22	1.6176
DisAttention	**0.2133±0.0049**	**0.924±0.160**	**0.1462±0.0042**	*0.1798±0.0150*	*188.81±8.59*	**18.92±1.64**	*1.5698±0.0194*
DisMix	*0.2442±0.0119*	*1.029±0.093*	*0.1558±0.0090*	**0.1675±0.0090**	**187.14±17.82**	*21.69±4.91*	**1.5470±0.0095**

Table 3: **Molecular and protein–ligand prediction performance.** The current values are results from the existing training recipe of up to 180 epochs. Values with ± denote the mean and SD across 3 seeds. We also experimented with TGT and GraphGPT, but their performance was poor at small model sizes, and they required long runtimes. These results are available in the Appendix.

In Table 3, both distance-driven models achieve lower mean errors than PlainGT and Graphormer on all six QM9 properties and on LP-PDBbind. DisMix achieves a Gap MAE of 187.14meV compared with Graphormer's 399.04meV, a reduction of 53.1%; LP-PDBbind RMSE also decreases from 1.6176 to 1.5470, a reduction of 4.4%.

Meanwhile, DisAttention achieves lower mean errors than DisMix on four of the six QM9 properties, supporting the view that effective node value propagation is possible by constructing mixing weights from geometry alone, without Q/K. DisMix's lower mean errors on LUMO, Gap, and LP-PDBbind suggest that the benefit of content adjustment may depend on how strongly the downstream task relies on node features and how rich those features are.

Over the first 1–20 epochs in Figure 3, the log–log decay rates γ_e of DisMix and DisAttention are 0.291 and 0.289, respectively, exceeding Graphormer's 0.043 and GraphGPT's 0.126. Within this observed interval, both models reduced error more steeply as training progressed, and DisAttention, which omits Q/K, exhibited a decay rate similar to that of DisMix. This suggests that distance-



(c) Log–log decay

Model	γ_t	γ_e
PlainGT	0.011	0.011
Graphormer	0.043	0.043
TGT	0.094	0.094
GraphGPT	0.126	0.126
DisAttention	*0.289*	*0.289*
DisMix	**0.291**	**0.291**

OLS: epochs 1–20. $\log E = a - \gamma \log x$. $\hat{t} = c_m e$ implies $\gamma_t = \gamma_e$.

Solid: source curves; dashed: descriptive fits. Unequal model sizes.

Figure 3: Learning curves and decay exponents for QM9 gap. Solid lines show validation MAE, and dashed lines show log–log OLS fits over the first 1–20 epochs. Lower error is better, and a larger γ indicates a steeper decrease within the corresponding interval.

defined mixing can serve as a useful structural constraint for early training progress as well as final error.

4.3 Beyond Molecular Graphs

We evaluate on MNIST superpixel (Monti et al., 2017) and ModelNet40 (Wu et al., 2015) to determine whether distance-driven mixing is also effective for non-molecular geometries. In Table 4, DisAttention achieves accuracies of $98.90 \pm 0.09\%$ and 83.47%, respectively. This shows that the same mixing structure can be used for spatial relations that are not specific to molecules. On ModelNet40, DisAttention's accuracy is 5.92 percentage points higher than that of DisMix, indicating that adding content affinity does not yield benefits for every geometry. This difference suggests that the relative advantages of the two operators may vary with the roles of geometry and node features.

Model	MNIST	ModelNet40
PlainGT	97.57±0.50	81.81
Graphormer	96.68±0.36	*82.50*
DisAttention	**98.90±0.09**	**83.47**
DisMix	*98.36±0.37*	77.55

Table 4: **Non-molecular graphs.** Test accuracy (%) ↑. MNIST reports the mean ± SD across 3 seeds; ModelNet40 uses a single seed.

4.4 Ablation of the Mixing Operator

To distinguish the roles of the learned radial bias, distance gate, and symmetric normalization on QM9 gap, we compare the five DisMix configurations in Table 5, with DisAttention and PlainGT as auxiliary controls.

Configuration	b	g	Norm	MAE (meV) ↓	Paired Δ ↓
PlainGT	–	–	–	787.98±0.63	731.85±0.98
Gaussian + content	–	–	✓	62.68±1.20	6.39±1.38
w/o radial bias	–	✓	✓	57.64±0.80	1.35±0.62
w/o gate	✓	–	✓	58.28±0.08	1.99±0.10
w/o normalization	✓	✓	–	*57.48±1.07*	*1.19±1.26*
DisAttention	✓	✓	✓	57.68±1.05	**0.79±0.44**
DisMix	✓	✓	✓	**56.29±0.18**	–

Table 5: Test MAE at the best validation checkpoint on QM9 gap (mean ± standard deviation).

DisMix achieves the lowest mean error at 56.29 ± 0.18meV and has lower error than every single-component removal condition for each of the two common seeds. Fixing the distance gate to $g(d) = 1$ increases the paired error by 1.99 ± 0.10meV, producing the largest mean performance degradation and the smallest variation across seeds among the single-component removals. This shows the benefit of a gate that adjusts distance-specific transmission strength after normalization.

Removing the radial bias and symmetric normalization also increases error by 1.35 ± 0.62 and 1.19 ± 1.26meV, respectively, indicating contributions from both distance-specific affinity adjustment and degree-based scaling. However, the magnitude of the normalization effect varies considerably even between the two seeds. Removing both the radial bias and the gate increases the error by 6.39 ± 1.38meV, supporting the design that jointly learns distance-specific affinity and transmission strength over using only a Gaussian prior and content.

DisAttention's paired difference is small at 0.79 ± 0.44meV, showing that distance-based mixing without Q/K achieves performance close to DisMix in this setting, while content adjustment provides an additional improvement. PlainGT's 787.98 ± 0.63meV is the result of a content-only control without distances, bonds, or positional encodings, illustrating the role of incorporating geometry into mixing under this training setup.

4.5 Learned Distance Responses and Mixing Patterns

We examine whether the learned distance functions treat all neighbors similarly or distinguish transmission paths according to the layer and surrounding geometry. Figure 4 compares the first-layer $W^{(1)}$ with the two-layer scalar paths $[W^{(2)}W^{(1)}]_{ij} = \sum_k W^{(2)}_{ik} W^{(1)}_{kj}$ for the same QM9 molecule. This product gives the weights of paths through intermediate nodes.

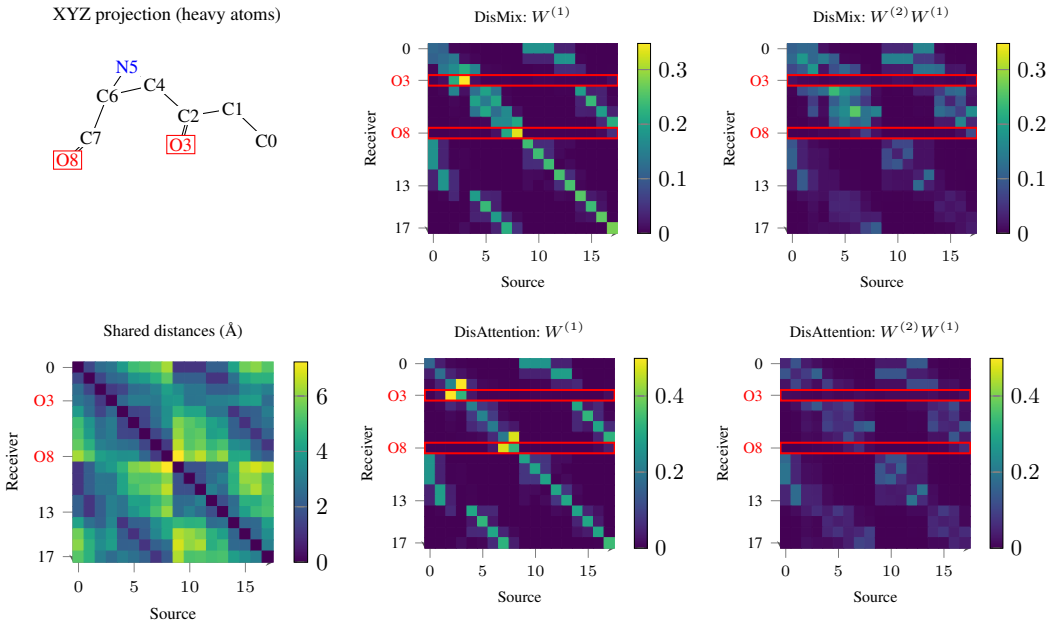


Figure 4: Structure and head0 mixing of QM9 molecule gdb_98735. The upper left shows a planar projection of the actual XYZ coordinates and the original bonds, with H omitted. The red boxes enclose the entire receiver rows of the two carbonyl oxygens, O3 and O8. In both models, these two oxygens have the largest relative decreases in row sums from $W^{(1)}$ to $W^{(2)}W^{(1)}$. DisMix's row sums decrease as $0.614 \rightarrow 0.110$ for O3 and $0.608 \rightarrow 0.203$ for O8, while in DisAttention, 21% and 24% of the original values remain, respectively.

For the two carbonyl oxygens O3 and O8 marked by red boxes, DisMix's receiver row sums decrease as $0.614 \rightarrow 0.110$ and $0.608 \rightarrow 0.203$, respectively. These two oxygens are also the receivers with the largest relative decreases in DisAttention. Thus, composing layers adjusts the total scalar path weight differently for each atom, and this redistribution also occurs in a distance-based operator without Q/K content scores. This observation differs from row-stochastic mixing, which keeps the total path weight at 1 for every receiver. Diagnostics for all 16 validation molecules and all layers/heads are provided in Appendix B.

Across the off-diagonal pairs of 16 molecules and 8 heads, the mean DisMix gate in the 4–8Å range changes from 0.6423 in layer 1 to 1.0569 in layer 8. The gating multiplier attenuates transmission

on average in this distance range in the first layer and approaches $g = 1$ in the last layer, indicating that even a fixed distance range is used differently across depths. The KL information relative to the uniform distribution, measured after normalizing the gates into probabilities over neighbors, also increases from 0.074bits in the first layer to 0.280bits in the last layer (Table 6). At approximately 11/21/41 epochs of the same run, the mean across all layers increases to $0.146/0.209/0.266$bits, showing how initially uniform gates acquire distance selectivity during training (Figure 7). Because the changes in intermediate layers are not monotonic, this observation indicates differentiation in distance use across layers rather than a uniform increase with depth.

In MolPCBA validation interventions examining whether these distance paths also contribute to actual predictions, neutralizing the radial bias and gates around the functional group produced a larger logit decrease than controls matched for atom count for all 16 positive molecules of the tertiary amine–assay combination selected through exploration, in both seeds (Appendix B.2). This can be viewed as evidence that the learned radial adjustments are used as paths supporting positive predictions.

5 Discussion and Limitations

Efficiency and scaling. The computational efficiency of our approach comes from using geometry as a more compact mixing operator. DisAttention removes Q/K projections and query–key comparisons, and neither model maintains or updates high-dimensional pair representations across layers. However, dense value mixing still requires $O(N^2 d)$ computation and $O(N^2 H)$ space to store head-specific weights. Here, N is the number of nodes, d is the hidden width, and H is the number of heads. Because the costs of radial functions and node updates also remain, the reduction in parameter count does not necessarily match the reduction in runtime or peak memory. The practical benefits of distance-driven mixing must therefore be assessed by considering not only model size but also execution costs under identical measurement conditions and the training cost required to reach a target level of performance.

Expressivity and the role of content. DisAttention learns how to pass information according to geometry, and DisMix adds relation adjustments based on node content. The relative performance of the two operators may vary with the roles of geometry and content required by the task. Mixing across multiple layers enables information to pass through intermediate nodes, but this does not guarantee that these pairwise operators alone can represent all higher-order geometric functions (Li et al., 2023; 2025). The trade-off between the efficiency of compact relational representations and the geometric expressivity required by the task is therefore important.

Future directions. Future extensions to sparse mixing could use learned distance responses to compute only the necessary pairs, alleviating the quadratic cost on large graphs. Architectures that compute content affinity only in the layers or heads where it is needed could also provide a way to select a data-appropriate performance–cost trade-off between DisAttention and DisMix. Such extensions would allocate computation according to graph size and task requirements while retaining the design of constructing mixing around geometry.

6 Conclusion

We proposed an approach that goes beyond using a distance matrix as an auxiliary attention bias by using it as the direct basis of an operator that mixes node values. DisAttention constructs distance-driven mixing without Q/K, while DisMix combines distance-based affinity with content bias to allow relations to be adjusted according to node states. Both models apply a distance gate after symmetric normalization to jointly control relative mixing proportions and message strength. The experiments demonstrate molecular prediction performance with compact models and applicability to non-molecular geometries, supporting the possibility of learning useful node interactions without repeatedly updating high-dimensional pair representations. These results show that using geometry itself as a learnable mixing operator can be a practical alternative for graph learning that pursues both performance and computational efficiency.

AI USE STATEMENT

During the preparation of this work, the authors used ChatGPT and Codex to refine the presentation of ideas and to assist with proof-of-concept (PoC) coding. Generative AI tools were strictly not used for generating the core ideas, establishing the research intent, or defining the scope of the experimental validation. All AI-assisted text and code were thoroughly cross-reviewed by the authors to ensure complete consistency with the actual implementation and empirical results. The authors assume full responsibility for the final content and artifacts presented in this paper.

ETHICS STATEMENT

This research utilizes widely adopted and publicly available datasets (e.g., QM9, MolPCBA, LP-PDBbind, and ModelNet40) that are permitted for academic research. The authors foresee no significant ethical concerns, potential for malicious misuse, or conflicts of interest associated with the methodologies and experimental results presented in this work.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our results, detailed descriptions of the model architecture, hyperparameters, and experimental settings are provided in Appendix C. Furthermore, the complete source code and data processing scripts necessary to replicate all experiments are available in an anonymous git repository: `https://anonymous.4open.science/r/iclr27gnn-publish-B616/README.md`.

REFERENCES

Guy W. Bemis and Mark A. Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of Medicinal Chemistry*, 39(15):2887–2893, 1996. doi: 10.1021/jm9602928. URL https://pubs.acs.org/doi/10.1021/jm9602928.

Zhe Chen, Hao Tan, Tao Wang, Tianrun Shen, Tong Lu, Qiuying Peng, Cheng Cheng, and Yue Qi. Graph propagation transformer for graph representation learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 3559–3567, 2023. doi: 10.24963/ijcai.2023/396. URL https://www.ijcai.org/proceedings/2023/396.

Xiaomin Fang, Lihang Liu, Jieqiong Lei, Donglong He, Shanzhuo Zhang, Jingbo Zhou, Fan Wang, Hua Wu, and Haifeng Wang. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence*, 4(2):127–134, 2022. doi: 10.1038/s42256-021-00438-4. URL https://www.nature.com/articles/s42256-021-00438-4.

Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1eWbxStPH.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1263–1272, 2017. URL https://proceedings.mlr.press/v70/gilmer17a.html.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2005.00687.

Weihua Hu, Matthias Fey, Hongyu Ren, Maho Nakata, Yuxiao Dong, and Jure Leskovec. OGB-LSC: A large-scale challenge for machine learning on graphs. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/db8e1af0cb3aca1ae2d0018624204529-Abstract-round2.html.

Md Shamim Hussain, Mohammed J. Zaki, and Dharmashankar Subramanian. Global self-attention as a replacement for graph convolution. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 655–665, 2022. doi: 10.1145/3534678.3539296. URL https://dl.acm.org/doi/10.1145/3534678.3539296.

Md Shamim Hussain, Mohammed J. Zaki, and Dharmashankar Subramanian. Triplet interaction improves graph transformers: Accurate molecular graph learning with triplet graph transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 20768–20792, 2024. URL https://proceedings.mlr.press/v235/hussain24a.html.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Jie Li, Xingyi Guan, Oufan Zhang, Kunyang Sun, Yingze Wang, Dorian Bagni, and Teresa Head-Gordon. Leak proof PDBBind: A reorganized data set of protein–ligand complexes for more generalizable binding affinity prediction. *The Journal of Physical Chemistry B*, 130(2):730–740, 2026. doi: 10.1021/acs.jpcb.5c08598. URL https://pubs.acs.org/doi/10.1021/acs.jpcb.5c08598.

Zian Li, Xiyuan Wang, Yinan Huang, and Muhan Zhang. Is distance matrix enough for geometric deep learning? In *Advances in Neural Information Processing Systems*, volume 36, pp. 37413–37447, 2023. doi: 10.52202/075280-1627. URL https://papers.neurips.cc/paper_files/paper/2023/hash/75f1a165c7561e028c41d42fa6286a76-Abstract-Conference.html.

Zian Li, Xiyuan Wang, Shijia Kang, and Muhan Zhang. On the completeness of invariant geometric deep learning models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=52x04chyQs.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Shuqi Lu, Zhifeng Gao, Di He, Linfeng Zhang, and Guolin Ke. Data-driven quantum chemical property prediction leveraging 3D conformations with Uni-Mol+. *Nature Communications*, 15:7104, 2024. doi: 10.1038/s41467-024-51321-w. URL https://www.nature.com/articles/s41467-024-51321-w.

Shengjie Luo, Tianlang Chen, Yixian Xu, Shuxin Zheng, Tie-Yan Liu, Liwei Wang, and Di He. One transformer can understand both 2D & 3D molecular data. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vZTp1oPV3PC.

Łukasz Maziarka, Tomasz Danel, Sławomir Mucha, Krzysztof Rataj, Jacek Tabor, and Stanisław Jastrzębski. Molecule attention transformer. *arXiv preprint arXiv:2002.08264*, 2020. doi: 10.48550/arXiv.2002.08264. URL https://arxiv.org/abs/2002.08264. Presented at the Graph Representation Learning and Machine Learning and the Physical Sciences workshops at NeurIPS 2019.

Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model CNNs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. doi: 10.1109/CVPR.2017.576. URL https://arxiv.org/abs/1611.08402.

Wonpyo Park, Woonggi Chang, Donggeon Lee, Juntae Kim, and Seung-won Hwang. GRPE: Relative positional encoding for graph transformer. In *ICLR Workshop on Machine Learning for Drug Discovery*, 2022. URL https://openreview.net/forum?id=GNfAFN_p1d.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. doi: 10.1109/CVPR.2017.16. URL https://arxiv.org/abs/1612.00593.

Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1: 140022, 2014. doi: 10.1038/sdata.2014.22. URL https://www.nature.com/articles/sdata201422.

Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 9323–9332, 2021. URL https://proceedings.mlr.press/v139/satorras21a.html.

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/303ed4c69846ab36c2904d3ba8573050-Abstract.html.

Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. Synthesizer: Rethinking self-attention for transformer models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 10183–10192, 2021. URL https://proceedings.mlr.press/v139/tay21a.html.

Philipp Thölke and Gianni De Fabritiis. Equivariant transformers for neural network based molecular potentials. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=zNHzqZ9wrRB.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. doi: 10.1109/CVPR.2015.7298801. URL https://arxiv.org/abs/1406.5670.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems*, volume 34, pp. 28877–28888, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f1c1592588411002af340cbaedd6fc33-Abstract.html.

Qifang Zhao, Weidong Ren, Tianyu Li, Hong Liu, Xingsheng He, and Xiaoxiao Xu. GraphGPT: Generative pre-trained graph Eulerian transformer. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 77630–77656, 2025. URL https://proceedings.mlr.press/v267/zhao25r.html.

Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-Mol: A universal 3D molecular representation learning framework. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6K2RM6wVqKu.

A COMPUTATIONAL EFFICIENCY

Differences in computational cost arise from the operations used to construct mixing weights and the representations maintained across layers. Let H be the number of heads and $d = Hd_h$ the total hidden width, and consider a dense block whose Q/K/V and output projections have total input and output widths of d. Excluding the shared FFN and radial function computations, the multiply–accumulate counts for the main matrix multiplications are

$$\begin{aligned} \mathrm{MAC}_{\mathrm{DisAttention}} &= 2Nd^2 + N^2 d, \\ \mathrm{MAC}_{\mathrm{DisMix}} &= 4Nd^2 + 2N^2 d. \end{aligned} \tag{7}$$

For DisAttention, $2Nd^2$ corresponds to the value and output projections, and $N^2 d$ to value mixing. DisMix adds Q/K projections and query–key score computation. Thus, at the same width, DisAttention removes $2Nd^2 + N^2 d$ of these main operations and, excluding projection biases, does not use $2d^2$ Q/K parameters per layer.

Let c_{rad} be the cost per node pair of evaluating the radial functions for all heads. Both models additionally require $O(N^2 c_{\mathrm{rad}})$ radial computations and $O(N^2 H)$ normalization and gating computations. The Q/K operations removed in equation 7 therefore do not directly correspond to the runtime savings for the entire block, and the quadratic complexity of dense value mixing remains. DisMix also retains the Q/K computations of standard attention but does not add high-dimensional pair-state updates to represent geometry.

Storing the head-specific mixing weights requires $O(N^2 H)$ space for both models, and only updated node representations are passed between layers. This differs from approaches that use d_p-dimensional pair representations, maintaining and updating $O(N^2 d_p)$ states. However, intermediate activations in the radial networks also consume memory, so we evaluate operator storage separately from actual peak memory. The current implementation computes the full dense matrix and does not exploit symmetry to compute or store only a subset of its entries. The computational efficiency therefore comes from removing Q/K in DisAttention and omitting high-dimensional pair-state updates in both models, rather than from computational savings due to symmetry itself. We evaluate the effects of these structural differences on actual training through training step time, peak memory, and wall-clock time to reach a target level of performance. Gaussian/RBF initialization and backbone details are provided in Appendix C.

B SUPPLEMENTARY MEASUREMENTS

B.1 DISTANCE RESPONSES AND GEOMETRY

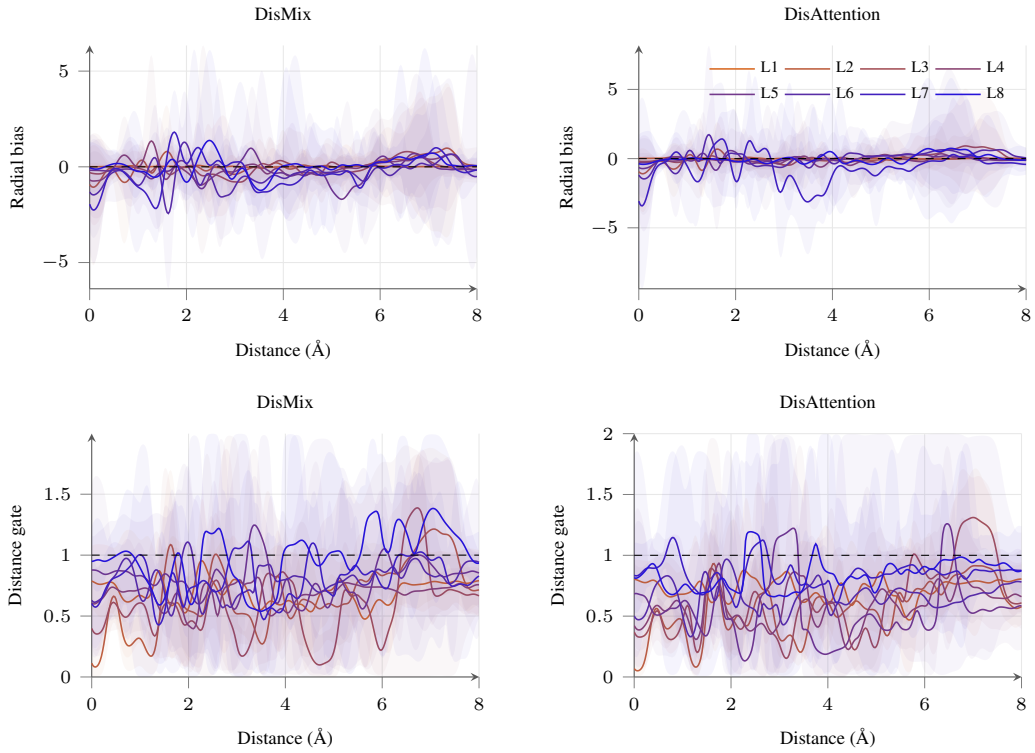


Figure 5: Responses from all layers/heads of the actual QM9 seed42 checkpoint. Radial biases and gates are shown on a common distance grid.

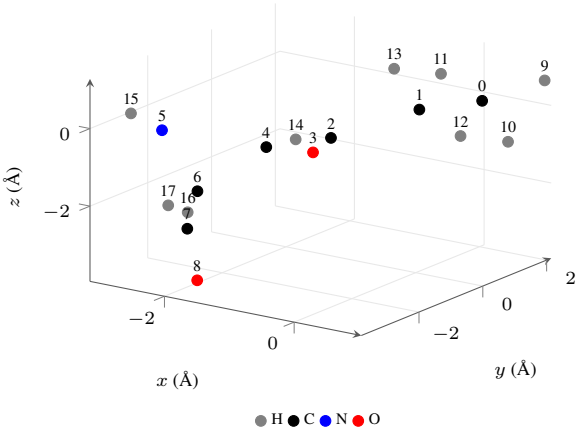


Figure 6: Atom ordering and XYZ coordinates of gdb_98735, selected using a fixed rule.

Layer-wise gate selectivity. We measure how selectively the saved gates distinguish distances to different neighbors across all 8 layers and 8 heads of the same checkpoint used in the figures above.

$$p_{ij}^{(\ell,h)} = \frac{g_{ij}^{(\ell,h)}}{\sum_{k\neq i} g_{ik}^{(\ell,h)}}, \qquad \bar{H}_i^{(\ell,h)} = \frac{-\sum_{j\neq i} p_{ij}^{(\ell,h)} \log_2 p_{ij}^{(\ell,h)}}{\log_2(N-1)},$$

$$I_i^{(\ell,h)} = D_{\mathrm{KL}}(p_i^{(\ell,h)} \| U_{N-1}) = \log_2(N-1)(1-\bar{H}_i^{(\ell,h)}), \qquad \mathrm{CV}_i^{(\ell,h)} = \frac{\mathrm{std}_{j\neq i}(g_{ij}^{(\ell,h)})}{\mathrm{mean}_{j\neq i}(g_{ij}^{(\ell,h)})}.$$

At initialization, all gates are 1, so $\bar{H} = 1$, $I = 0$, and $\mathrm{CV} = 0$. That is, a uniform distribution over neighbors has maximum Shannon entropy and contains no information distinguishing distances. Selectivity after training is reflected in a decrease in $\bar{H}$ and an increase in information I relative to the uniform distribution.

	DisMix			DisAttention		
Layer	$\bar{H}\downarrow$	I (bits) $\uparrow$	CV $\uparrow$	$\bar{H}\downarrow$	I (bits) $\uparrow$	CV $\uparrow$
Init. $g=1$	1.000	0.000	0.000	1.000	0.000	0.000
1	0.981	0.074±0.006	0.285	0.984	0.061±0.006	0.263
2	0.953	0.175±0.026	0.420	0.969	0.111±0.028	0.365
3	*0.915*	*0.340±0.046*	*0.643*	0.949	0.203±0.029	0.492
4	0.918	0.325±0.039	0.559	0.905	0.370±0.038	0.648
5	0.941	0.236±0.033	0.514	**0.768**	**0.920±0.114**	**1.073**
6	0.944	0.217±0.021	0.473	0.917	0.323±0.023	0.618
7	**0.874**	**0.479±0.039**	**0.763**	0.945	0.210±0.019	0.517
8	0.928	0.280±0.020	0.561	*0.883*	*0.446±0.046*	*0.711*

Table 6: Layer-wise gate selectivity for 16 fixed, length-stratified validation molecules. Values were averaged over receivers and heads within each molecule, then weighted equally across molecules.

DisMix's I increases from 0.074 bits in layer 1 to 0.280 bits in layer 8, and CV also increases from 0.285 to 0.561. The most selective layer is layer 7 ($I = 0.479$bits). For DisAttention, this measure also increases from 0.061 bits to 0.446 bits, but its maximum is 0.920 bits in layer 5. For both models, the mean I in the last layer exceeds that in the first layer for each of the 16 molecules. However, changes in intermediate layers are not monotonic, supporting the interpretation that learned layer-wise selectivity forms and differentiates from initially uniform gates.

Changes during training. We applied the same aggregation procedure to the same 16 molecules at saved checkpoints from the same run. The mean I across all layers of DisMix increases to 0.146/0.209/0.266 bits at approximately 11.34/21.42/41.49 epochs, respectively, while that of DisAttention increases to 0.183/0.238/0.330 bits at 9.95/17.00/40.30 epochs. Across the three observed time points, the mean I increased in all 8 layers of both models. Thus, separately from the non-monotonic differentiation across depths, we observe initially uniform gates acquiring distance selectivity as training progresses.

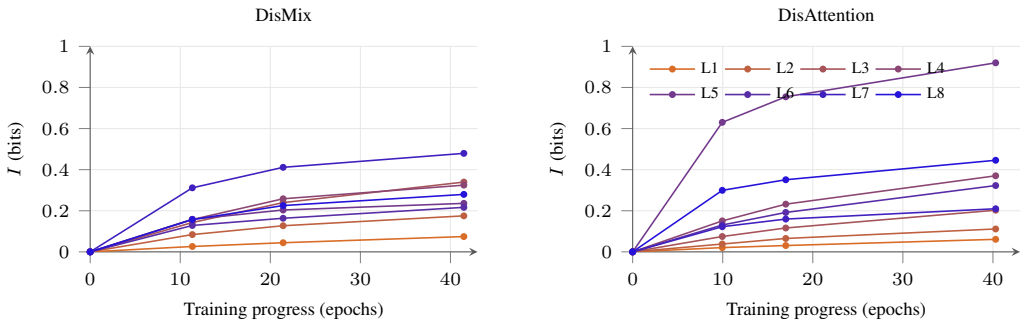


Figure 7: Layer-wise gate information computed at saved validation-best milestone checkpoints from the same seed42 run. Epoch is the number of completed epochs plus the fraction of batches processed in the current epoch. Epoch0 is the reference point given by the initialization $g = 1$.

B.2 MOLPCBA FUNCTIONAL-GROUP PROBES

We explored which functional groups have surrounding learned distance paths on which the positive predictions of 13M DisMix depend. We predefined 16 SMARTS functional groups and divided the validation molecules into two sets using hashes of Murcko scaffolds (Bemis & Murcko, 1996). Among 104 functional-group–assay combinations that met the label-count, positive-enrichment, and AP criteria in the exploration set, we investigated 11 after reducing redundancy. After fixing the three candidates with the largest positive logit decreases relative to controls in seed42, we checked them in a separate scaffold set and checked the same molecules with seed2026. In each set, we selected correctly classified positive and negative molecules based on the saved seed42 logits, up to 16 of each, in ID hash order.

Neutralizing distance paths. For off-diagonal pairs with a functional-group atom as one endpoint, we set the learned radial biases and gates in all 12 layers to $b = 0, g = 1$. Coordinates, atom types, bonds, the Gaussian prior, and the content pathway are retained, while normalization is recomputed. For each molecule, we also compute four controls by selecting the same number of atoms outside the functional group. We define the difference in task logits before and after neutralization as $\Delta z = z - z_{\text{neutral}}$ and compare $\Delta z_{\text{excess}} = \Delta z_{\text{motif}} - \frac{1}{4}\sum_{r=1}^{4} \Delta z_{\text{control},r}$, which subtracts the control mean. This is a diagnostic intervention on learned radial paths in a fixed molecule, rather than an atom-deletion or conformer-change experiment. All interventions were computed using FP32 inference with a fixed validation-best EMA model.

			Mean excess $\Delta z \uparrow$		Excess $> 0 \uparrow$	
Motif	AID	n	Seed42	Seed2026	Seed42	Seed2026
Tertiary amine	720532	16	*4.59*	**4.25**	**16/16**	**16/16**
Urea	504847	11	**4.66**	*1.59*	**11/11**	*8/11*
Phenol	588579	16	0.32	0.34	*9/16*	7/16

Table 7: Results on the separate scaffold set for the three candidates fixed after exploration. Both seeds use the same positively labeled molecules. AID is the PubChem assay ID in the official OGB source.

The most consistent candidate. For tertiary amine–AID720532, the excess was positive for all 16 molecules in both seeds. The mean logit decreases were 5.314 for the functional group and 0.719 for controls in seed42, and 4.931/0.682 in seed2026. The mean positive scores $\sigma(z)$ decreased from 0.961 to 0.472 and from 0.963 to 0.430, respectively. The figure below shows the molecule ranked 8th among the 16 by score decrease in seed42, rather than the case with the largest decrease.

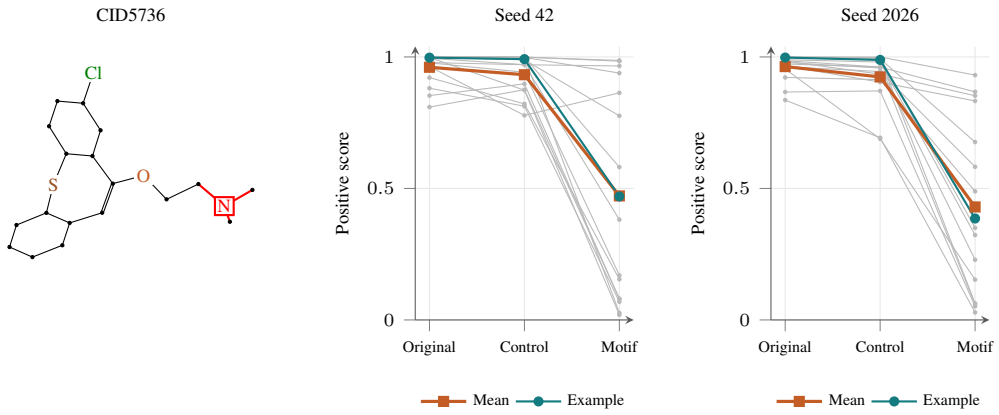


Figure 8: The actual input conformer of the tertiary amine candidate and all 16 positive molecules. The left panel shows a PCA projection of XYZ with equal axis scaling. The red N and three bonds mark the tertiary amine location, and all off-diagonal pairs incident to that N are neutralized. Dots denote C, lines denote the original bonds, and H is omitted. Each gray line on the right represents one molecule. Original is the original score, Control is the mean score of the four controls, and Motif is the score after neutralizing the functional-group paths.

Interpretation and limitations. Logits also decreased for 16 negatively labeled molecules containing the same functional group, but the mean excess was 1.898/1.370 for seeds42/2026, smaller than the 4.595/4.250 for positive molecules. This is consistent with dependence on distance paths within the molecular context, rather than an explanation in which the presence of the functional group alone determines a positive prediction. Urea also showed positive mean excess in both seeds, but the magnitude varied from 4.662 to 1.594. The effect of phenol was small, and in seed2026 the mean score instead increased after functional-group neutralization. Thus, the most consistent observation is that learned radial adjustments around tertiary amines support positive scores.

C Experimental Details

Experiments were conducted using NVIDIA RTX5090 GPUs, with NVIDIA GH200 used only in cases requiring large amounts of VRAM. We confirmed that errors arising from computational differences between the two chips were within the range of seed-induced variation. We used seeds 1, 42, and 2026.

QM9 operator comparison. For QM9 (Ramakrishnan et al., 2014), we use the existing training/validation/test split of 110,000/10,000/10,831 examples, with coordinates in Å. Only the training data were used for target normalization.

Distance function implementation. The Gaussian log-prior for head h is $-(d-\mu_h)^2/w_h$, where $w_h = \mathrm{softplus}(r_h) + 0.1$. The centers are initialized using a fixed random seed $7717 + \ell$ for each layer to sample from a uniform distribution on $[0.5, 4.5]$ and sort the samples. Each initial effective width is 1. The centers and the untransformed width tensors are learned. The two independent radial MLPs use 64 fixed RBF centers equally spaced over $[0, 8]$. The basis functions are $\exp[-\gamma(d-c_k)^2]$, with $\gamma = (63/8)^2$. Each MLP has a $64 \to 64 \to H$ architecture with GELU. Both output layers are initialized to 0, so initially $b(d) = 0$ and $g(d) = 2\sigma(q(d)) = 1$. The end of the basis interval does not imply a cutoff that forcibly removes connections based on distance. The diagonal entries of the symmetrized content score are 0, and the head-specific learnable coefficients $\tanh(\beta_h)$ start at 0. The QM9 backbone uses 8 blocks, a width of 256, 8 heads, and an FFN width of 1024. These are followed by LayerNorm, masked max pooling, and a $256 \to 128 \to 1$ prediction head. Because DisAttention has no active Q/K projections, it has fewer parameters even at the same width and depth.

MolPCBA model. The small model has 12 blocks, a width of 384, 12 heads, and 13,018,400 parameters. We transfer 343 backbone tensors from the epoch 2 checkpoint of our Affinity pretraining on PCQM4Mv2 (Hu et al., 2021)[1] and newly initialize the head for 128 tasks and the optimizer. We use mean pooling, an FFN that preserves width, and residual scaling, with dropout/attention-dropout/maximum drop-path rates of 0.1. The content coefficient of the pretrained model is initialized to 0.95. This coefficient and the learned distance filters are retained at the start of downstream training. We use a logical batch size of 128, AdamW (Loshchilov & Hutter, 2019) with a learning rate of 3×10^{-4} and weight decay of 10^{-5}, and gradient clipping at 5. The existing 100-epoch schedule uses 5 epochs of warmup followed by cosine decay to 10^{-6}. Early stopping has a patience of 25 and is applied after at least 40 epochs. All three downstream seeds use the same fixed pretrained initialization and the PyTorch 2.8 / BF16 environment on Pizza. The EMA decay rate is 0.999, and evaluation uses a single EMA network. After excluding 349 molecules for which conformer generation failed from the existing scaffold split (Hu et al., 2020), the 3D cohort consists of 350,227/43,696/43,657 training/validation/test molecules. We predict 128 tasks, of which 127 allow computation of mean AP. Our pretraining excludes 41 examples from the official PCQM4Mv2 training SDF and uses an internal training/validation split of 3,209,637/168,928 examples.

ModelNet40. For ModelNet40 (Wu et al., 2015), we center the existing 2048-point clouds in PointNet format (Qi et al., 2017) and normalize them to a maximum radius of 1, then select 1024 points using deterministic farthest-point selection. The starting point is the point farthest from the origin, with ties broken by the original index order. XYZ is provided both as coordinates and as three numerical node features. The shared categorical embedding is constant ($z = 0$, mapped to embedding index 1), but the full node input is not constant. Normal vectors are not used. We use 8,856 training shapes, a stratified validation sample of 984 shapes, and the official 2,468 test shapes. We do not apply rotation or jitter augmentation or test voting, and use masked max pooling. The reported ModelNet40 accuracy is an observation from a single seed=1.

Additional QM9 results. Because the performance of the SOTA models TGT and GraphGPT was not fully realized under constraints on model size and training time, the main text compares only models with similar sizes and training speeds.

Model	QM9 Gap MAE (meV) ↓	number of seeds
PlainGT	793.43±0.58	3
Graphormer	399.04±2.17	3
DisAttention	*188.81±8.59*	3
DisMix	**187.14±17.82**	3
TGT	597.95	1
GraphGPT	554.86	1

Number of training epochs The QM9 cost efficiency experiments used results from training for up to 180 epochs. The QM9 ablation study used results from training for up to 45 epochs with two seeds, 1 and 42.

[1]PCQM4Mv2 version and training-structure documentation: `https://ogb.stanford.edu/docs/lsc/pcqm4mv2/`.