COOPHYPERMOE: MEMBERSHIP INTERPOLATION OVER AN EXPERT HYPERGRAPH FOR MOLECULAR GRAPH PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph mixture-of-experts (MoE) models adapt message passing to heterogeneous graph inputs, yet a top-K router must select experts even when every routing score is weak. We introduce CoopHyperMoE, a membership-based router that keeps node–expert affinities independent before selection. It periodically builds a hypergraph from similarities between routing vectors and, for low-membership inputs, interpolates non-anchor memberships from principal-component anchors in the active hyperedges. A normalized top-K gate is then computed from the revised memberships. We evaluate CoopHyperMoE with four fixed seeds and validation-selected checkpoints on ten molecular benchmarks: seven classification datasets and three regression datasets. Without a virtual node, CoopHyperMoE records a higher mean ROC-AUC than a locally reproduced GMoE on five of seven classification tasks and raises the seven-task average from 71.90 to 72.04; it also obtains the lowest local RMSE on Lipophilicity. Matched ablations on six classification datasets test interpolation, annealing, and the activation rule. A final-checkpoint audit finds no interpolation triggers on the BACE and Tox21 validation splits, limiting what these comparisons establish about the interpolation mechanism.

1 INTRODUCTION

Molecular property prediction requires models to learn from graphs whose local chemical environments and global structures can vary substantially. Message-passing graph neural networks (GNNs) provide a natural framework by iteratively aggregating information from atoms and bonds (Kipf & Welling, 2017; Gilmer et al., 2017; Xu et al., 2019). Pretraining and self-supervised learning further improve molecular representations by exploiting unlabeled graphs and informative learning objectives (Hu et al., 2020b; Rong et al., 2020; Xia et al., 2023). These advances, however, leave a complementary architectural question: how should a model allocate its transformations across heterogeneous molecular inputs? Even with informative node representations, a shared message-passing transformation must accommodate diverse local environments, motivating input-dependent combinations of specialized transformations.

Mixture-of-experts (MoE) architectures address this question through multiple experts and learned routing (Shazeer et al., 2017; Fedus et al., 2022). For graph learning, GMoE routes nodes to aggregation experts with different receptive fields (Wang et al., 2023), TopExpert incorporates molecular topology into expert assignment (Kim et al., 2023), and Mowst coordinates weak and strong predictors through a confidence-dependent mechanism (Zeng et al., 2024). More broadly, recent routing studies explore adaptive expert activation in DynMoE (Guo et al., 2025) and continuous ReLU routing in ReMoE (Wang et al., 2025). These developments show that routing design involves more than increasing the expert count: the representation and use of routing scores also determine how expert capacity is combined.

A remaining issue for conventional softmax top-K routing is the distinction between *relative preference* and *independently computed membership*. Top-K selection always returns the prescribed number of experts, while softmax allocates a fixed amount of probability mass among the candidates. In particular, adding the same constant to every routing logit leaves both the ranking and the softmax weights unchanged. The resulting gate therefore does not retain information about the

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.

common level of the logits. Independent sigmoid memberships preserve this information before normalization and allow several memberships to be simultaneously high or low. Prior work has also studied the statistical advantages of sigmoid gating in specific MoE regression settings (Nguyen et al., 2024). Here, we use memberships as internal routing signals for conditional refinement; their magnitudes are not calibrated probabilities of expert correctness.

Preserving memberships raises a second question: how should the router revise them when none exceeds its activation threshold? Independent computation alone does not specify how evidence from related experts should influence an assignment. We investigate a relational approach in which groups of experts provide a structure for membership interpolation before final selection. A hypergraph can represent overlapping groups involving more than two experts, allowing an expert to participate in multiple interpolation neighborhoods. We construct these groups from routing-vector similarity, using the learned router geometry as a parameter-based prior for refinement. This choice defines an operational relation among experts without assuming that parameter similarity establishes functional complementarity.

We introduce **CoopHyperMoE**, a molecular graph MoE model that combines independent membership computation with conditional interpolation over an expert hypergraph. At each message-passing layer, the router computes a sigmoid membership for every node–expert pair. A periodically refreshed hypergraph organizes experts by routing-vector similarity, and principal-component analysis selects anchors within each group. For low-membership nodes with eligible hyperedges, Gaussian interpolation transfers anchor memberships to non-anchor experts, with an interpolation strength scheduled across layers and training. The revised memberships then determine the normalized top-K gate. This design makes expert relations intervene at the score-refinement stage, before the final expert combination is formed.

We evaluate CoopHyperMoE on ten molecular benchmarks using four fixed seeds and validation-selected checkpoints, and examine its components through matched ablations on six classification tasks. The results show modest, task-dependent gains over locally reproduced GMoE, while the ablations do not establish a consistent benefit from conditional interpolation. Routing diagnostics further distinguish the presence of the interpolation mechanism from its actual activation. Our contributions are: **(1) A membership-based routing formulation.** We separate independent node–expert memberships from normalized top-K selection, making the pre-normalization scores available to a conditional refinement mechanism. **(2) Expert-hypergraph membership interpolation.** We combine overlapping expert groups, principal-component anchors, and scheduled Gaussian membership interpolation to revise routing scores before selection. **(3) An empirical assessment of the routing design.** We report molecular classification and regression comparisons, matched component ablations, and activation diagnostics to examine both predictive behavior and the conditions under which interpolation is used.

2 Related Work

Graph MoE and routing. Mixtures of local experts learn input-dependent combinations (Jacobs et al., 1991); sparse MoE uses conditional activation to scale capacity (Shazeer et al., 2017; Fedus et al., 2022). BASE and Expert Choice constrain assignments (Lewis et al., 2021; Zhou et al., 2022). Graph-specific designs include receptive-field routing in GMoE (Wang et al., 2023), diverse graph classifiers in GraphDIVE (Hu et al., 2022), topology-informed experts in TopExpert (Kim et al., 2023), adaptive-depth experts in DA-MoE (Yao et al., 2024), and subgraph experts in MoSE (Ye et al., 2025). Mowst mixes weak and strong predictors (Zeng et al., 2024), while MoG studies graph sparsification (Zhang et al., 2025). ReMoE uses a differentiable ReLU router (Wang et al., 2025); our independent sigmoid memberships instead precede a discontinuous top-K gate. Published results from these models are not protocol-matched to ours.

Molecular prediction and expert relations. Molecular predictors use message passing, scaffold-aware evaluation, pretraining, and large-scale transformers (Gilmer et al., 2017; Yang et al., 2019; Hu et al., 2020b; Rong et al., 2020). Splits, features, and checkpoint rules therefore matter when comparing routers. Load balancing can discourage routing collapse without proving that experts have distinct functions (Shazeer et al., 2017; Lewis et al., 2021). Hypergraphs encode relations among more than two entities (Zhou et al., 2006; Feng et al., 2019); ours groups experts by router-

weight similarity, not by measured output complementarity. Whether grouped experts learn complementary functions remains untested.

3 METHOD

Overview. Figure 1 follows one node through a MoE layer. Each expert receives the same graph, while the router computes independent node–expert memberships. A hypergraph built from routing-vector similarities groups experts and is reused across nodes between refreshes.
For a low-membership node, an eligible group supplies PC anchors for conditional interpolation; otherwise, the original memberships pass through. The resulting scores define a normalized top-K gate before graph pooling and prediction.

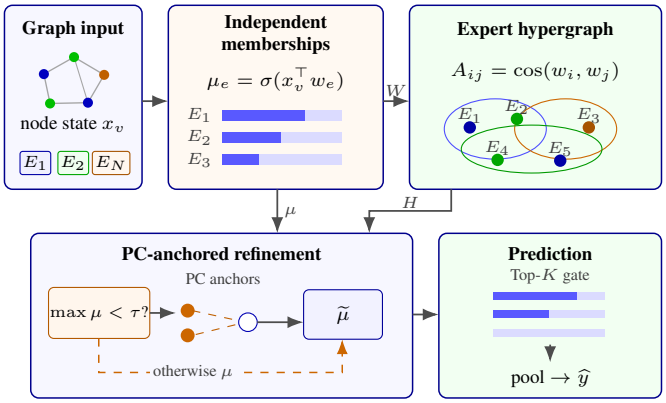

Figure 1: CoopHyperMoE routing pipeline.

3.1 PROBLEM FORMULATION AND COMPUTATIONAL UNIT

Let $G = (V, E, X)$ denote a molecular graph with node and edge features. A stack of L message-passing layers produces node states $x_v^{(\ell)}$, with $\ell \in \{0, \ldots, L-1\}$ indexing the MoE layers. Graph pooling and a prediction head map the final states to graph-level outputs. For classification, y contains one or more binary labels and the outputs are task logits. In each MoE layer, several message-passing experts transform the same input graph. The router operates on node states, so membership testing and interpolation are node-conditional even though supervision is graph-level. Nodes within the same molecule can consequently receive different expert combinations.

We use N for the expert count, K for the final gate cardinality, and M for the requested number of hyperedges. Layer superscripts are omitted when the operation is local to a single layer. The virtual-node version changes the underlying graph representation and is treated as a separate backbone.

3.2 INDEPENDENT MEMBERSHIPS

For node v's current representation $x_v \in \mathbb{R}^d$, a learned matrix $W = [w_1, \ldots, w_N] \in \mathbb{R}^{d \times N}$ defines independent memberships

$$\mu_{ve} = \sigma(x_v^\top w_e), \qquad e = 1, \ldots, N. \tag{1}$$

Unlike softmax, the sigmoid does not couple distinct logits at this stage: $\partial\sigma(z_i)/\partial z_j = 0$ for $i \neq j$. Several experts may therefore have simultaneously high or low memberships without sharing a fixed normalization budget. These values serve as internal signals for the refinement rule, rather than calibrated probabilities of expert correctness.

For the same logits and a common tie-breaking rule, sigmoid and softmax preserve the same ordering and hence yield the same top-K set before interpolation. Independent memberships instead preserve score magnitudes for thresholding and relational refinement, and can yield different normalized mixture weights. Interpolation can subsequently change the ordering before selection. Interpolation, top-K, and final normalization reintroduce cross-expert dependence; Appendix A states this scope precisely.

3.3 A PERIODICALLY REFRESHED EXPERT HYPERGRAPH

The expert-hypergraph block in Figure 1 groups experts, not input nodes. The columns of W define routing vectors. In contrast to hypergraph learning over data instances (Zhou et al., 2006; Feng et al.,

2019), our vertices are experts and the relation structure is refreshed from router parameters. At a fixed refresh interval, pairwise cosine similarities are computed as

$$C_{ij} = \frac{w_i^\top w_j}{\|w_i\|_2 \|w_j\|_2}, \qquad C_{ii} = -1. \tag{2}$$

Cosine similarity measures alignment between routing directions while removing their norm dependence. It provides a parameter-based criterion for grouping experts whose gates respond to related directions in representation space.

A greedy procedure uses highly similar pairs as hyperedge seeds and may add a third expert with positive mean similarity to the pair. For hyperedges $\mathcal{E}_m$, the binary incidence matrix and normalized edge weights are

$$H_{em} = \mathbf{1}[e \in \mathcal{E}_m], \qquad \rho_m = \frac{q_m}{\sum_{r=1}^{M} q_r + \epsilon}, \quad q_m \geq 0, \tag{3}$$

where q_m is the nonnegative seed-pair affinity or the singleton fallback weight. Overlapping hyperedges allow an expert to participate in several interpolation neighborhoods, rather than requiring a disjoint partition of the expert pool. Each group supplies a joint anchor set, and its contribution is resolved through the node-dependent aggregation defined below. In the experimental configuration, $N = 8$, $M = 4$, and the structure is refreshed every 200 training forward passes.

Periodic refresh lets the grouping track the evolving router while reusing the cached structure between updates. The grouping operation is not differentiated. Since the groups are derived from routing parameters rather than measured expert outputs, they define a prior for membership interpolation without establishing functional complementarity. No separate cooperation loss is imposed on the hypergraph.

3.4 Principal-component anchors and membership interpolation

The refinement stage constructs reference memberships within expert groups and applies them only when both the node-level trigger and edge eligibility conditions hold. This separates the construction of interpolation targets from the decision to use them.

Expert attributes and anchor selection. For expert e at layer ℓ, the implementation concatenates its routing vector, the mean norm of eligible convolution-weight tensors, and normalized depth:

$$f_e^{(\ell)} = \left[w_e^{(\ell)} \,\middle\|\, \frac{1}{|\Theta_e|} \sum_{\theta \in \Theta_e} \|\theta\|_F \,\middle\|\, \frac{\ell}{L-1} \right]. \tag{4}$$

Here Θ_e is the set of convolution-weight tensors included in the summary. The routing coordinates and parameter-norm statistic define an attribute space for selecting and comparing reference experts; they do not measure predictive accuracy.

Within hyperedge m, let the row-centered feature matrix satisfy $\widetilde{F}_m = U_m \Sigma_m V_m^\top$. The explained-variance fraction of the first component and the number of anchors are

$$r_{1,m} = \frac{\sigma_{m,1}^2}{\sum_j \sigma_{m,j}^2 + \epsilon}, \qquad n_{a,m} = \begin{cases} |\mathcal{E}_m|, & |\mathcal{E}_m| \leq 2, \\ \max\{2, \lfloor |\mathcal{E}_m|(1 - r_{1,m}) \rfloor\}, & |\mathcal{E}_m| > 2. \end{cases} \tag{5}$$

Writing $\widetilde{f}_e$ for the centered feature row and $v_{m,1}$ for the leading principal direction, the $n_{a,m}$ experts with the smallest absolute projections $|\widetilde{f}_e^\top v_{m,1}|$ form the anchor set A_m. This rule favors reference members near the center along the dominant direction of within-group variation. It selects centrally projected references rather than maximally separated extremes. For groups of at most two experts, every member is an anchor and no non-anchor target is needed.

Interpolation within and across groups. For a non-anchor $e \in \mathcal{E}_m \setminus A_m$, the edge-specific Gaussian interpolation target is

$$p_{ve}^{(m)} = \sum_{a \in A_m} \alpha_{ea}^{(m)} \mu_{va}, \qquad \alpha_{ea}^{(m)} = \frac{\exp[-\|f_e - f_a\|_2^2/(2\sigma^2)]}{\sum_{b \in A_m} \exp[-\|f_e - f_b\|_2^2/(2\sigma^2)]}. \tag{6}$$

The coefficients assign greater weight to anchors closer in expert-attribute space. Since they are nonnegative and sum to one, $p_{ve}^{(m)}$ lies between the smallest and largest anchor memberships for node v. Thus the target redistributes existing membership evidence within the group. The kernel compares expert attributes and interpolates memberships, not expert predictions. Its bandwidth σ is fixed by the run configuration.

An edge is active for node v when its maximum original membership exceeds τ_{edge}. Writing $b_{vm} = \mathbf{1}[\max_{j:H_{jm}=1} \mu_{vj} > \tau_{\text{edge}}]$, overlapping active-edge targets are aggregated as

$$\beta_{vem} = \frac{b_{vm}\rho_m H_{em}}{\sum_{r=1}^{M} b_{vr}\rho_r H_{er} + \epsilon}, \qquad \bar{p}_{ve} = \sum_{m=1}^{M} \beta_{vem}\, t_{ve}^{(m)}, \tag{7}$$

where $t_{ve}^{(m)} = \mu_{ve}$ for an anchor and $p_{ve}^{(m)}$ for a non-anchor. Anchor status is local to a hyperedge: an expert can retain its original membership as a target in one group while receiving an interpolated target in another. Equation 7 combines these contributions using the eligible groups containing that expert.

Conditional activation and scheduled blending. The adjustment is enabled only when $\max_e \mu_{ve} < \tau_{\text{active}}$ and the expert belongs to at least one active edge with positive weight:

$$\begin{aligned} \mu'_{ve} &= \text{clip}_{[0,1]}(\mu_{ve} + m_{ve}\lambda_{\ell,t}(\bar{p}_{ve} - \mu_{ve}))\,, \\ m_{ve} &= \mathbf{1}\left[\max_j \mu_{vj} < \tau_{\text{active}}\right] \mathbf{1}\left[\sum_m b_{vm}\rho_m H_{em} > 0\right], \end{aligned} \tag{8}$$

with the spatiotemporal schedule

$$\lambda_{\ell,t} = \lambda_0 \left(1 - \frac{\ell}{L-1}\right) \frac{1 + \cos(\pi t/T)}{2}, \tag{9}$$

where $t \in [0, T]$ denotes progress over the prescribed training horizon. The depth factor assigns greater interpolation strength to earlier layers, while the cosine factor reduces that strength during training. In particular, the schedule gives $\lambda_{L-1,t} = 0$ and $\lambda_{\ell,T} = 0$. The annealed quantity is the mixing strength, not the Gaussian bandwidth. When $m_{ve} = 0$, the original membership is retained.

The two thresholds serve distinct roles: τ_{active} identifies low-membership nodes, whereas τ_{edge} requires some membership support within a candidate group. A necessary condition for an enabled adjustment is therefore $\tau_{\text{edge}} < \max_e \mu_{ve} < \tau_{\text{active}}$, requiring $\tau_{\text{edge}} < \tau_{\text{active}}$. Nodes whose memberships all fall below the edge threshold have no eligible group. Even when the mask is active, a nonzero displacement requires both a nonzero mixing strength and an aggregated target different from the original membership. The rule consequently defines a conditional score correction, not a guaranteed out-of-distribution fallback.

3.5 Sparse gating and training objective

After refinement, let $S_v = \text{TopK}(\mu'_v, K)$. The final gate is

$$g_{ve} = \frac{\mathbf{1}[e \in S_v]\mu'_{ve}}{\sum_{j \in S_v} \mu'_{vj} + \epsilon}. \tag{10}$$

Applying selection after refinement allows the revised memberships to affect both the selected set and the relative weights within it. A membership change need not change the selected set, since the modified score may remain on the same side of the top-K boundary.

The evaluated implementation computes each expert output $E_e^{(\ell)}(x_v, G)$ and returns

$$x_v^{(\ell+1)} = \frac{1}{N} \sum_{e=1}^{N} g_{ve}^{(\ell)} E_e^{(\ell)}(x_v^{(\ell)}, G). \tag{11}$$

Zeros in g remove mixture weights but do not skip expert computation. The N^{-1} factor in Eq. 11 also means that the output is not a conventional convex combination. This factor is retained to match the evaluated implementation; the current model does not provide sparse-execution speedup.

For labeled molecular classification targets, the task term is binary cross entropy with logits; missing task labels are masked. At layer ℓ, define expert importance $I_e^{(\ell)} = \sum_v g_{ve}^{(\ell)}$, hard load $D_e^{(\ell)} = \sum_v \mathbf{1}[g_{ve}^{(\ell)} > 0]$, and $\mathrm{CV}^2(a) = \mathrm{Var}(a)/(\mathrm{Mean}(a)^2 + \epsilon)$. The implemented objective is

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \frac{1}{L} \sum_{\ell=0}^{L-1} \left\{ \gamma \Big[\mathrm{CV}^2(I^{(\ell)}) + \mathrm{CV}^2(D^{(\ell)}) \Big] - \eta \, \frac{1}{|V|} \sum_v \max_e \mu_{ve}^{\prime(\ell)} \right\}, \tag{12}$$

with $\gamma = \eta = 0.01$ in all reported runs. Importance summarizes the total gate mass assigned to each expert, whereas hard load counts its nonzero assignments. The negative maximum-membership term favors a larger strongest membership for each node; it does not directly reward agreement or complementarity between expert outputs. The virtual-node variant is reported separately because it changes the backbone.

Each forward pass computes memberships, refreshes expert groups and anchors when scheduled, applies conditional interpolation and top-K, and combines the expert outputs. Algorithms 1–2 in Appendix B provide the corresponding implementation steps. Grouping, hard-load counting, and top-K selection remain discrete operations.

4 Experiments

4.1 Experimental protocol and classification results

We use OGB scaffold splits and the OGB evaluator (Hu et al., 2020a). As emphasized by MoleculeNet and subsequent molecular benchmarks (Wu et al., 2018; Yang et al., 2019), the split and metric are part of the evaluation protocol. The seven classification datasets use ROC-AUC, and the three regression datasets use RMSE. Every run uses five GNN layers, hidden width 150, dropout 0.5, 150 epochs, batch size 32, Adam with learning rate 10^{-3}, eight experts, a top-4 gate, and training-only node drop of 0.2. The four seeds are $\{0, 42, 100, 2026\}$. We report the test score at the best validation epoch and compute an aggregate only when all four seeds completed.

The comparison unit is a dataset–backbone–method–seed tuple. We separate GCN, GCN+virtual node (VN), local GMoE, and the two CoopHyperMoE backbones. Published results with different splits, features, or tuning budgets are contextual rather than protocol-matched baselines.

The local rows in Table 1 are calculated from saved per-seed logs. Every local cell contains four completed runs with 150 evaluations each. TopExpert and the GraphDIVE re-evaluation use the GCN results in TopExpert's Table 1 (Kim et al., 2023); the latter must not be confused with GraphDIVE's original three-dataset results (Hu et al., 2022). DA-MoE uses the reported GCN variant in its Table 2 (Yao et al., 2024). These external results are not protocol-matched and are excluded from local rankings and aggregates. The OGB HIV leaderboard reference appears separately in Appendix Table 6. Runs from different local batches do not isolate a single module effect. BACE was not among the ten tasks in the original GMoE study (Wang et al., 2023).

4.2 Molecular regression comparison

Our experiments are limited to graph-level molecular prediction. ESOL, FreeSolv, and Lipophilicity provide three four-seed regression comparisons (Wang et al., 2023); node, link, and distribution-shift tasks require separate protocols.

Table 1: Molecular classification ROC-AUC (%, mean ± sample SD); SD is set in smaller type to the right of the mean. Local rows use four fixed seeds and validation-selected checkpoints under the OGB protocol. External rows (*) are contextual, not protocol-matched; their provenance is specified below. Avg.-7 and boldface compare completed local rows only; "–" means unavailable or not calculated here.

Method	HIV	Tox21	BACE	BBBP	ToxCast	ClinTox	SIDER	Avg.-7
Local Reproductions (4-seed OGB protocol)								
GCN	$72.66_{\pm 0.98}$	$73.67_{\pm 0.51}$	$77.13_{\pm 1.64}$	$68.74_{\pm 0.66}$	$63.53_{\pm 0.55}$	$\mathbf{90.43}_{\pm 1.57}$	$\mathbf{59.16}_{\pm 0.33}$	**72.19**
GCN+VN	$74.04_{\pm 0.44}$	$\mathbf{76.03}_{\pm 0.48}$	$77.61_{\pm 1.00}$	$67.89_{\pm 1.51}$	$\mathbf{65.58}_{\pm 0.42}$	–	–	–
GMoE	$\mathbf{76.51}_{\pm 1.12}$	$74.18_{\pm 0.87}$	$76.64_{\pm 1.88}$	$68.15_{\pm 0.42}$	$61.52_{\pm 0.73}$	$89.64_{\pm 1.27}$	$56.66_{\pm 1.76}$	71.90
Published Reports (GCN backbone; different protocols)								
GraphDIVE*	$65.20_{\pm 4.10}$	$73.20_{\pm 0.90}$	$59.40_{\pm 4.60}$	$62.50_{\pm 1.90}$	$62.00_{\pm 0.70}$	$60.40_{\pm 6.20}$	$51.90_{\pm 2.00}$	–
TopExpert*	$76.60_{\pm 0.70}$	$75.30_{\pm 0.40}$	$76.40_{\pm 1.90}$	$67.00_{\pm 2.40}$	$63.10_{\pm 0.60}$	$70.10_{\pm 6.00}$	$60.80_{\pm 0.90}$	–
DA-MoE*	$77.62_{\pm 1.58}$	$75.59_{\pm 0.69}$	–	$69.62_{\pm 0.98}$	$65.18_{\pm 0.48}$	–	–	–
CoopHyperMoE (Ours, 4-seed OGB protocol)								
Ours	$75.22_{\pm 0.70}$	$74.93_{\pm 0.23}$	$\mathbf{78.32}_{\pm 1.33}$	$68.79_{\pm 1.55}$	$61.93_{\pm 0.48}$	$85.95_{\pm 1.91}$	$59.13_{\pm 1.18}$	72.04
Ours+VN	$75.93_{\pm 1.13}$	$74.71_{\pm 0.66}$	$77.32_{\pm 4.96}$	$\mathbf{68.91}_{\pm 1.48}$	$63.05_{\pm 0.89}$	–	–	–

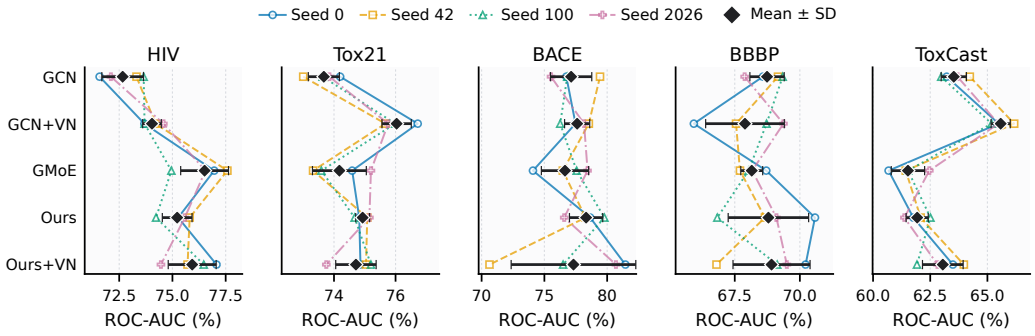

Figure 2: Four-seed OGB classification results from Table 1. Colored lines join results with the same seed across methods; they indicate pairing, not a continuous path. Black diamonds and bars denote means and sample standard deviations ($n = 4$). Horizontal scales differ by dataset. ClinTox and SIDER are reported in the table and Appendix C.1.

The local regression results have no single winner (Table 2 and Figure 3). GCN has the lowest mean RMSE on ESOL, GMoE on FreeSolv, and CoopHyperMoE on Lipo. In every task, the gap between the two lowest means is smaller than the sample SD of at least one of those methods. Thus, these four-seed rankings are descriptive rather than evidence of a reliable advantage. On FreeSolv, CoopHyperMoE has the largest across-seed variation of the three local methods; its mean alone gives an incomplete picture.

For literature context, the GCN-based DA-MoE report gives ESOL 1.13 ± 0.04 and FreeSolv 2.19 ± 0.07 RMSE (Yao et al., 2024). Those ten-run published values are not matched to our four-seed setting. The local datasets were run in separately provisioned environments, so their runtimes are not directly comparable.

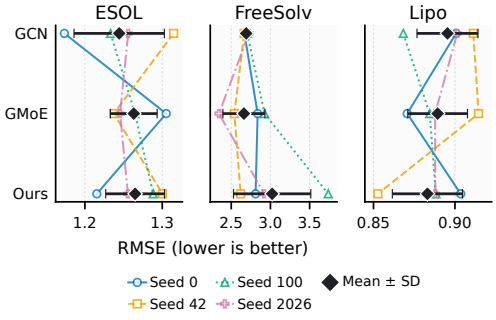

Figure 3: Four-seed regression RMSE. Colored lines connect matched seeds across methods; diamonds and bars give mean ± sample SD ($n = 4$). Lower is better; horizontal scales differ.

Table 2: Four-seed molecular-regression RMSE (mean ± sample SD; lower is better). Each value is validation-selected under its dataset-specific local environment.

Dataset	GCN	GMoE (local)	CoopHyperMoE–GCN
ESOL	1.244 ± 0.058	1.263 ± 0.030	1.265 ± 0.038
FreeSolv	2.691 ± 0.030	2.661 ± 0.268	3.022 ± 0.493
Lipo	0.895 ± 0.019	0.889 ± 0.018	$\mathbf{0.883 \pm 0.022}$

4.3 Ablation Study

We fix the split, backbone, seeds, optimizer, and checkpoint rule, then compare the full model with three variants: no interpolation, constant interpolation without annealing, and interpolation triggered for every input. A singleton hyperedge produces zero displacement, so it does not define a separate "no-hypergraph" variant. There is no cooperation-loss ablation because the model contains no such loss.

Table 3: Matched four-seed ablation ROC-AUC (%, mean ± sample SD). Every row uses a fixed backbone, source hash, checkpoint rule, and seed set. Bold marks the largest mean within each dataset.

Dataset	Full model	No interpolation	No annealing	Always trigger
BACE	77.63 ± 1.47	78.63 ± 1.28	$\mathbf{79.43 \pm 3.09}$	78.65 ± 0.87
ToxCast	62.50 ± 0.72	$\mathbf{62.84 \pm 0.97}$	61.79 ± 1.25	61.99 ± 0.90
BBBP	67.08 ± 1.24	68.69 ± 1.35	69.30 ± 2.01	$\mathbf{69.51 \pm 0.86}$
ClinTox	85.95 ± 1.91	85.24 ± 3.02	84.68 ± 2.36	$\mathbf{86.25 \pm 2.37}$
SIDER	59.13 ± 1.18	58.96 ± 1.28	$\mathbf{60.13 \pm 1.25}$	59.43 ± 1.01
Tox21	74.60 ± 0.59	$\mathbf{75.00 \pm 0.41}$	74.87 ± 0.37	74.46 ± 0.14

On BACE and BBBP, the full mechanism is numerically below all three simplified variants (Table 3 and Figure 4). Disabling annealing gives the highest BACE mean but also its largest dispersion; always triggering gives the highest BBBP mean. The BACE seed trajectories are mixed: without annealing, seeds 42 and 2026 improve over the full model, while seeds 0 and 100 decline. The 1.80-point mean increase is accompanied by a rise in sample SD from 1.47 to 3.09 points, rather than a uniform gain across runs. On ToxCast and Tox21, removing interpolation gives the highest mean. Always triggering is highest on ClinTox (86.25 ± 2.37), while disabling annealing is highest on SIDER (60.13 ± 1.25). The full model has the highest mean on none of these six datasets. No significance test is reported; the ablation does not establish a consistent benefit from conditional, annealed interpolation.

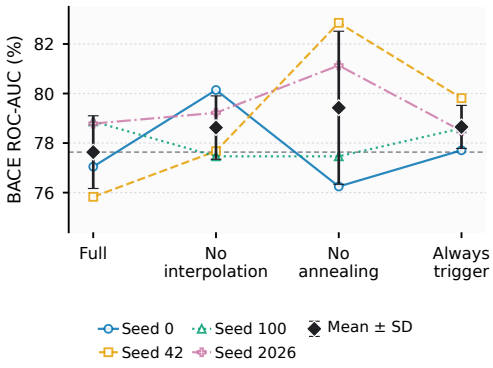

Figure 4: Matched BACE ablation. Lines connect matched seeds; diamonds and bars give mean ± sample SD ($n = 4$). The dashed line marks the full-model mean; the vertical axis is truncated.

4.4 Analysis and Discussion

Appendix D examines routing at selected checkpoints and along one interpolation path. A new validation-set probe of the four final epoch-150 checkpoints finds no interpolation triggers in BACE (112,440 node–layer evaluations) or Tox21 (419,100); the final predictions on these splits therefore use the unchanged memberships. This finding is limited to the final checkpoints, not the training

trajectory, and the ablation differences cannot be attributed to active interpolation at inference on these splits. Agreement between hyperedges and expert outputs remains untested.

A one-seed BACE screen varies interpolation strength, trigger threshold, refresh interval, and K (Table 11 in Appendix C). The reference setting gives the highest validation ROC-AUC (72.31), narrowly ahead of $\tau_{\text{active}} = 0.7$ (72.23). Test scores are shown only for completeness and were not used to choose a setting.

Table 4: Matched training cost on ClinTox (RTX 4090 D, seed 0). All models use five GCN layers, width 150, batch size 32, and 150 epochs. Time is the synchronized mean over training epochs 6–150; validation and checkpoint I/O are excluded. Peak memory is PyTorch allocated memory over the run.

Method	Trainable params.	Peak memory (MiB)	Train time/epoch (s)
GCN	151,652	38.0	0.85
GMoE	1,040,402	152.1	2.82
CoopHyperMoE	1,034,402	152.2	4.06

Table 4 shows similar parameter counts and peak allocated memory for GMoE and CoopHyperMoE, but CoopHyperMoE takes about 44% longer per training epoch on this workload. All experts are evaluated in the current implementation, so the top-K gate does not reduce expert computation.

No configuration dominates all seven classification tasks (Table 1). Non-VN CoopHyperMoE exceeds local GMoE on five tasks but trails on HIV and ClinTox; its Avg.-7 is 72.04, versus 71.90 for GMoE and 72.19 for GCN. The VN variant exceeds GCN+VN on HIV and BBBP but trails it on Tox21, BACE, and ToxCast.

CoopHyperMoE reaches $78.32 \pm 1.33\%$ on BACE and $68.79 \pm 1.55\%$ on BBBP, above the listed TopExpert means, but falls below both local GMoE and TopExpert on HIV. DA-MoE reports higher means on its four overlapping classification tasks. These cross-paper differences are not protocol-matched: features, splits, seeds, and tuning budgets may differ. Virtual nodes can change both the magnitude and direction of differences, and the 4.96-point BACE dispersion for CoopHyperMoE+VN cautions against attributing cross-backbone gaps to routing alone.

The classification gains are therefore modest and task-dependent. The regression results likewise show no consistent advantage. Together, the evidence supports a competitive router on some molecular tasks, not a general improvement over the baselines.

The ablations narrow the interpretation further. A simplified variant has the highest mean on every ablated dataset, and the final-checkpoint probes on BACE and Tox21 find no active interpolation on their validation splits. These probes do not measure whether interpolation occurred earlier in training; conversely, the observed score differences cannot establish that interpolation caused a gain. Establishing the value of the proposed correction will require trigger counts over training and matched comparisons that isolate routing from the other implementation changes.

5 Conclusion

We introduced CoopHyperMoE, which combines independent sigmoid memberships, a periodically refreshed expert hypergraph, and conditional membership interpolation before top-K selection. The implementation is evaluated with four seeds on ten molecular benchmarks, with matched ablations on six classification datasets. Across the seven classification tasks, the non-VN model records a higher mean ROC-AUC than local GMoE on five and improves its average from 71.90 to 72.04; on the three regression tasks, it gives the best local result on Lipophilicity. Simpler routing variants are stronger on individual datasets, and the final-checkpoint audit finds no interpolation triggers on two validation splits. These observations do not establish that the interpolation mechanism causes the reported gains. Further limitations include cross-batch comparisons in part of the evaluation, discontinuous top-K selection, dense evaluation of all experts, and no cooperation objective for the hypergraph. Measuring trigger activity during training and testing expert complementarity are important next steps.

AI USE STATEMENT

Generative AI tools assisted with manuscript drafting and copyediting, literature organization, experimental planning, and consistency checks against source code and saved logs. They did not supply or generate missing training results. The authors are responsible for the validity of every claim, citation, and reported measurement.

REPRODUCIBILITY STATEMENT

Section 3 defines the evaluated router, and Section 4 specifies the datasets, splits, metrics, seeds, and checkpoint rule. The appendix provides implementation pseudocode (Section B), run settings and aggregation rules (Section C), per-seed results (Section C.1), and the scope of the routing diagnostics (Section D). Reported local aggregates were calculated from saved per-seed logs. An anonymized code-and-log package is planned but is not claimed to be available with this manuscript.

REFERENCES

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3558–3565, 2019. doi: 10.1609/aaai.v33i01.33013558.

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 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 2017.

Yongxin Guo, Zhenglin Cheng, Xiaoying Tang, Zhaopeng Tu, and Tao Lin. Dynamic mixture of experts: An auto-tuning approach for efficient transformer models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c63ff34c163bca3055d39490f6d1aaf7-Abstract-Conference.html.

Fenyu Hu, Liping Wang, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Graphdive: Graph classification by mixture of diverse experts. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 2080–2086, 2022. doi: 10.24963/ijcai.2022/289.

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*, 2020a.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020b. URL https://openreview.net/forum?id=HJlWWJSFDH.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79.

Tianyi Jiang, Zeyu Wang, Shanqing Yu, and Qi Xuan. Adaptive substructure-aware expert model for molecular property prediction, 2025. URL https://arxiv.org/abs/2504.05844.

Suyeon Kim, Dongha Lee, SeongKu Kang, Seonghyeon Lee, and Hwanjo Yu. Learning topology-specific experts for molecular property prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8291–8299, 2023. doi: 10.1609/aaai.v37i7.26000.

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.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. BASE layers: Simplifying training of large, sparse models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6265–6274. PMLR, 2021.

Huy Nguyen, Nhat Ho, and Alessandro Rinaldo. Sigmoid gating is more sample efficient than softmax gating in mixture of experts. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://papers.nips.cc/paper_files/paper/2024/hash/d65befe6b80ecf7f180b4def503d7776-Abstract-Conference.html.

Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Yan Sun, Yutong Lu, Yan Yi Li, Zihao Jing, Carson K. Leung, and Pingzhao Hu. Molgraph-xlstm as a graph-based dual-level xlstm framework for enhanced molecular representation and interpretability. *Communications Chemistry*, 8:286, 2025. doi: 10.1038/s42004-025-01683-z.

Haotao Wang, Ziyu Jiang, Yuning You, Yan Han, Gaowen Liu, Jayanth Srinivasa, Ramana Rao Kompella, and Zhangyang Wang. Graph mixture of experts: Learning on large-scale graphs with explicit diversity modeling. In *Advances in Neural Information Processing Systems*, 2023.

Ziteng Wang, Jun Zhu, and Jianfei Chen. Remoe: Fully differentiable mixture-of-experts with relu routing. In *International Conference on Learning Representations*, 2025.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. Moleculenet: A benchmark for molecular machine learning. *Chemical Science*, 9:513–530, 2018. doi: 10.1039/C7SC02664A.

Jun Xia, Chengshuai Zhao, Bozhen Hu, Zhangyang Gao, Cheng Tan, Yue Liu, Siyuan Li, and Stan Z. Li. Mole-BERT: Rethinking pre-training graph neural networks for molecules. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=jevY-DtiZTR.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Veronika Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2019. doi: 10.1021/acs.jcim.9b00237.

Zelin Yao, Chuang Liu, Xianke Meng, Yibing Zhan, Jia Wu, Shirui Pan, and Wenbin Hu. Damoe: Addressing depth-sensitivity in graph-level analysis through mixture of experts, 2024. URL https://arxiv.org/abs/2411.03025.

Junda Ye, Zhongbao Zhang, Li Sun, and Siqiang Luo. Mose: Unveiling structural patterns in graphs via mixture of subgraph experts, 2025. URL https://arxiv.org/abs/2509.09337.

Hanqing Zeng, Hanjia Lyu, Diyi Hu, Yinglong Xia, and Jiebo Luo. Mixture of weak and strong experts on graphs. In *International Conference on Learning Representations*, 2024.

Guibin Zhang, Xiangguo Sun, Yanwei Yue, Chonghe Jiang, Kun Wang, Tianlong Chen, and Shirui Pan. Graph sparsification via mixture of graphs. In *International Conference on Learning Representations*, 2025.

Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems*, volume 19, 2006.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M. Dai, Zhifeng Chen, Quoc V. Le, and James Laudon. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

A Scope of the routing-level derivative

For softmax logits $z_1, \ldots, z_N$ and probabilities $s_i = \exp(z_i)/\sum_k \exp(z_k)$, direct differentiation yields

$$\frac{\partial s_i}{\partial z_j} = s_i(\mathbf{1}[i = j] - s_j). \tag{13}$$

Thus the off-diagonal derivative is $-s_i s_j$ whenever $i \neq j$. For independent sigmoid memberships $\mu_i = \sigma(z_i)$, the Jacobian before any other processing is diagonal:

$$\frac{\partial \mu_i}{\partial z_j} = \mathbf{1}[i = j]\mu_i(1 - \mu_i). \tag{14}$$

These identities are algebraic facts about two score maps; they do not establish smoother predictions or lower training variance for the complete graph model. If the selected top-K set S is locally fixed, $g_i = \mu_i/(\sum_{k\in S} \mu_k + \epsilon)$ for $i \in S$, so a selected gate still has nonzero cross-dependence through its denominator. At a top-K boundary, the selected set can change abruptly. Any empirical claim about stable routing therefore needs a perturbation experiment on complete model outputs and selected-expert sets, not just the diagonal membership Jacobian.

Let $Z_S = \sum_{k\in S} \mu'_k + \epsilon$. Away from a support-change boundary, the normalized gate Jacobian is

$$\frac{\partial g_i}{\partial \mu'_j} = \mathbf{1}[i \in S]\left(\frac{\mathbf{1}[i = j]}{Z_S} - \frac{\mu'_i \mathbf{1}[j \in S]}{Z_S^2}\right). \tag{15}$$

Thus independent pre-interpolation memberships do not imply independent normalized gates. The derivative is valid only while S remains fixed; it does not describe the discontinuity introduced by exchanging the Kth and $(K + 1)$th experts.

B Implementation pseudocode

Algorithms 1 and 2 give the two procedures behind the equations in Section 3. They follow the current code path, including dense expert evaluation after sparse gate construction.

Algorithm 1 Periodic expert-hypergraph refresh and PC-anchor caching

Require: router matrix $W = [w_1, \dots, w_N]$, expert attributes $\{f_e\}_{e=1}^{N}$, edge budget M
Ensure: incidence H, normalized weights ρ, cached anchors $\{A_m\}$ and non-anchors $\{Q_m\}$

$C_{ij} \leftarrow w_i^\top w_j / (\|w_i\|_2 \|w_j\|_2)$; set $C_{ii} \leftarrow -1$
$H \leftarrow 0, q \leftarrow 0, m \leftarrow 1, \mathcal{P} \leftarrow \varnothing$
for candidate pairs (i, j) in descending order of C_{ij} **do**
  if $m > M$ **then**
    break
  end if
  if $i = j$ or $\{i, j\} \in \mathcal{P}$ **then**
    continue
  end if
  $\mathcal{E}_m \leftarrow \{i, j\}$
  $k \leftarrow \arg\max_{e \notin \{i,j\}} (C_{ie} + C_{je})/2$
  if $(C_{ik} + C_{jk})/2 > 0$ **then**
    $\mathcal{E}_m \leftarrow \mathcal{E}_m \cup \{k\}$
  end if
  $H_{em} \leftarrow \mathbf{1}[e \in \mathcal{E}_m]$; $q_m \leftarrow \max(C_{ij}, 0)$
  $\mathcal{P} \leftarrow \mathcal{P} \cup \{\{i, j\}\}$; $m \leftarrow m + 1$
end for
while $m \le M$ **do** ▷ singleton fallback for an unfilled column
  choose $e = (m - 1) \bmod N$; $H_{em} \leftarrow 1$; $q_m \leftarrow 1$; $m \leftarrow m + 1$
end while
$\rho_m \leftarrow q_m / (\sum_r q_r + \epsilon)$ for every edge m
for $m = 1, \dots, M$ **do**
  center the rows of $F_m = [f_e : e \in \mathcal{E}_m]$ and compute $F_m = U_m \Sigma_m V_m^\top$
  compute $r_{1,m}$ and $n_{a,m}$ from Eq. 5
  $A_m \leftarrow$ the $n_{a,m}$ members with smallest $|\widetilde{f}_e^\top v_{m,1}|$
  $Q_m \leftarrow \mathcal{E}_m \setminus A_m$
end for
return $H, \rho, \{A_m, Q_m\}_{m=1}^{M}$

Algorithm 2 Conditional membership interpolation and sparse gating at one MoE layer

Require: node states X, cached (H, ρ, A, Q), experts $\{E_e\}$, K, τ_{active}, τ_{edge}, $\lambda_{\ell,t}$, σ
Ensure: layer output Y and auxiliary loss $\mathcal{L}_{\text{aux}}^{(\ell)}$

$\mu \leftarrow \sigma(XW)$
for $m = 1, \dots, M$ **do**
  $b_{vm} \leftarrow \mathbf{1}[\max_{e:H_{em}=1} \mu_{ve} > \tau_{\text{edge}}]$ for all nodes v
  for $a \in A_m$ **do**
    $t_{va}^{(m)} \leftarrow \mu_{va}$ for all v
  end for
  for $e \in Q_m$ **do**
    $\alpha_{ea}^{(m)} \leftarrow \text{softmax}_{a \in A_m}(-\|f_e - f_a\|_2^2 / (2\sigma^2))$
    $t_{ve}^{(m)} \leftarrow \sum_{a \in A_m} \alpha_{ea}^{(m)} \mu_{va}$ for all v
  end for
end for
compute β_{vem} and $\bar{p}_{ve}$ using Eq. 7
$m_{ve} \leftarrow \mathbf{1}[\max_j \mu_{vj} < \tau_{\text{active}}] \mathbf{1}[\sum_m b_{vm} \rho_m H_{em} > 0]$
$\mu'_{ve} \leftarrow \text{clip}_{[0,1]}(\mu_{ve} + m_{ve} \lambda_{\ell,t} (\bar{p}_{ve} - \mu_{ve}))$
$S_v \leftarrow \text{TopK}(\mu'_v, K)$ and compute g_{ve} using Eq. 10
$I_e \leftarrow \sum_v g_{ve}$; $D_e \leftarrow \sum_v \mathbf{1}[g_{ve} > 0]$
$\mathcal{L}_{\text{aux}}^{(\ell)} \leftarrow \gamma[\text{CV}^2(I) + \text{CV}^2(D)] - \eta |V|^{-1} \sum_v \max_e \mu'_{ve}$
for $e = 1, \dots, N$ **do**
  compute $Z_e \leftarrow E_e(X, G)$ ▷ all experts are evaluated
end for
$Y \leftarrow N^{-1} \sum_{e=1}^{N} g_e \odot Z_e$
return $Y, \mathcal{L}_{\text{aux}}^{(\ell)}$

C IMPLEMENTATION AND EVALUATION CHECKLIST

Table 5: Settings used for the reported molecular experiments.

GNN layers	5	Hidden dimension	150
Dropout	0.5	Training epochs	150
Batch size	32	Adam learning rate	10^{-3}
Weight decay	0	Training node drop	0.2
Experts N	8	Selected gates K	4
Hyperedges M	4	Refresh interval	200 forwards
One-hop experts	8	Balance coefficient	0.01
τ_{active}	0.5	τ_{edge}	0.3
Blend λ_0	0.5	Kernel bandwidth σ	1.0
Seeds	$0, 42, 100, 2026$	Selection	Best validation epoch
Data split	OGB scaffold	Molecular metric	ROC-AUC

For exact reproduction, the experiment manifest records the OGB and PyTorch versions, graph features, virtual-node setting, Git revisions of the model and runner, task-specific masking rules, hardware, and per-seed log paths. The four non-VN HIV logs are stored with their commands and source hashes. Runs with different queue configurations are treated as different experiments even when they share a model name.

Command template. The following is a reproducibly equivalent non-VN CoopHyperMoE command template; replace the dataset and seed for each recorded tuple. It states the node-drop rate and weight decay explicitly, although the saved commands relied on those runner defaults.

```
python main_pyg.py --gnn gcn-fhmoe --dataset ogbg-moltoxcast \
  --seed 0 --num_layer 5 --emb_dim 150 --drop_ratio 0.5 \
  --epochs 150 --batch_size 32 --num_workers 4 --lr 0.001 \
  --drop_node_ratio 0.2 --weight_decay 0 \
  --num_experts 8 --num_experts_1hop 8 -k 4 --coef 0.01 \
  --tau_active 0.5 --tau_edge 0.3 --lambda_fri 0.5 \
  --n_hyperedges 4 --hyper_refresh 200 --sigma 1.0
```

Aggregate calculation example. The four CoopHyperMoE–GCN+VN ToxCast test scores are 63.4875, 63.9657, 61.9225, and 62.8169. Their mean and sample standard deviation round to 63.05 ± 0.89, as reported in Table 1.

For non-VN HIV, the four validation-selected CoopHyperMoE test percentages (seeds $0, 42, 100, 2026$) are 75.2740, 75.8124, 74.2264, and 75.5824. Their mean and sample standard deviation round to 75.22 ± 0.70. Seed 100 was resumed from a saved checkpoint; its two result fragments contain 81 and 69 evaluations, respectively. All four runs have 150 evaluations and successful exit codes in the archived September 15 log snapshot.

Checkpoint selection and result aggregation. For each dataset–method–seed run, the checkpoint is chosen using the validation metric only; the corresponding test metric is then used for the reported seed result. For a complete four-seed cell with test values $r_1, \ldots, r_4$, the reported mean and sample standard deviation are

$$\bar{r} = \frac{1}{4}\sum_{s=1}^{4} r_s, \qquad s_r = \sqrt{\frac{1}{3}\sum_{s=1}^{4}(r_s - \bar{r})^2}. \tag{16}$$

ROC-AUC is reported in percentage points (larger is better), whereas regression RMSE remains in each dataset's target units (smaller is better). Avg.-7 in Table 1 is the unweighted arithmetic mean of the seven completed classification dataset means, not a pooled ROC-AUC over molecules or tasks. We do not average classification with regression metrics, impute incomplete virtual-node rows, or include published, non-matched numbers in local aggregates. The detailed seed tables below expose the values underlying the corresponding means.

The OGB leaderboard reports the `ogbg-molhiv` test ROC-AUC entries in Table 6; values are converted from fractions to percentages. The leaderboard GCN mean exceeds our non-VN HIV

mean, while its virtual-node mean is close to ours. Sharing the dataset, split, and metric does not match architecture, optimization, seed count, or checkpoint selection; the leaderboard does not supply the other six classification tasks in Table 1.

Table 6: OGB leaderboard test ROC-AUC (%) for `ogbg-molhiv` only. These published GCN references use different model settings and are not included in local rankings or averages.

OGB leaderboard model	HIV
GCN	76.06 ± 0.97
GCN+virtual node	75.99 ± 1.19

C.1 Full per-seed evaluation breakdown

Table 7 gives per-seed test ROC-AUC for the first five classification datasets, and Table 9 covers the two completed later. Table 10 lists the regression runs. Every local aggregate uses seeds $\{0, 42, 100, 2026\}$ and the test score from the epoch with the best validation score. Table 8 separately records two further molecular-MoE literature reports.

Table 7: Detailed local per-seed breakdown and mean performance across the five archival OGB molecular classification datasets (ROC-AUC %). Each seed result is evaluated at the validation-selected optimal epoch. Subscripts denote sample standard deviations ($n = 4$). Δ indicates the average improvement over plain GCN. Literature results are reported separately in Tables 1 and 8; ClinTox and SIDER appear in Table 9.

Method	Seed	HIV	Tox21	BACE	BBBP	ToxCast	Avg.
GCN	0	71.58	74.20	76.75	68.58	63.22	70.87
	42	73.31	73.00	79.43	69.16	64.23	71.82
	100	73.65	73.63	76.82	69.34	62.98	71.28
	2026	72.11	73.85	75.53	67.88	63.69	70.61
	Mean	$72.66_{\pm 0.98}$	$73.67_{\pm 0.51}$	$77.13_{\pm 1.64}$	$68.74_{\pm 0.66}$	$63.53_{\pm 0.55}$	$\mathbf{71.15}_{\pm 0.53}$
GCN+VN	0	73.68	76.72	77.48	65.93	65.49	71.86
	42	74.21	75.67	78.54	67.55	66.17	72.43
	100	73.69	75.99	76.26	68.72	65.18	71.97
	2026	74.58	75.76	78.16	69.37	65.48	72.67
	Mean	$74.04_{\pm 0.44}$	$\mathbf{76.03}_{\pm 0.48}$	$77.61_{\pm 1.00}$	$67.89_{\pm 1.51}$	$\mathbf{65.58}_{\pm 0.42}$	$\mathbf{72.23}_{\pm 0.38\ (+1.08)}$
GMoE	0	76.96	74.59	74.09	68.71	60.68	71.00
	42	77.58	73.32	76.47	67.70	61.43	71.30
	100	74.95	73.59	77.57	68.03	61.53	71.13
	2026	76.55	75.20	78.44	68.17	62.46	72.16
	Mean	$\mathbf{76.51}_{\pm 1.12}$	$74.18_{\pm 0.87}$	$76.64_{\pm 1.88}$	$68.15_{\pm 0.42}$	$61.52_{\pm 0.73}$	$\mathbf{71.40}_{\pm 0.52\ (+0.25)}$
Ours	0	75.27	74.82	78.65	70.58	61.76	72.22
	42	75.81	75.08	78.28	68.66	62.08	71.98
	100	74.23	74.67	79.78	66.83	62.51	71.60
	2026	75.58	75.15	76.58	69.10	61.37	71.56
	Mean	$75.22_{\pm 0.70}$	$74.93_{\pm 0.23}$	$\mathbf{78.32}_{\pm 1.33}$	$68.79_{\pm 1.55}$	$61.93_{\pm 0.48}$	$\mathbf{71.84}_{\pm 0.30\ (+0.69)}$
Ours+VN	0	77.07	74.86	81.45	70.22	63.49	73.42
	42	75.71	75.04	70.63	66.80	63.97	70.43
	100	76.47	75.20	76.49	69.13	61.92	71.84
	2026	74.46	73.75	80.70	69.50	62.82	72.24
	Mean	$75.93_{\pm 1.13}$	$74.71_{\pm 0.66}$	$77.32_{\pm 4.96}$	$\mathbf{68.91}_{\pm 1.48}$	$63.05_{\pm 0.89}$	$\mathbf{71.98}_{\pm 1.24\ (+0.83)}$

Table 8: Additional molecular-MoE literature results on the five dataset names in Table 7 (reported test ROC-AUC %, mean ± SD). These are authors' reported scores, not reruns or matched baselines; "–" indicates that the cited paper does not report that dataset. No across-paper average or ranking is computed.

Published model	HIV	Tox21	BACE	BBBP	ToxCast
ASE-Mol (GCN) (Jiang et al., 2025)	85.5 ± 1.1	77.6 ± 0.8	96.3 ± 1.2	77.5 ± 1.8	73.1 ± 0.7
MolGraph-xLSTM (Sun et al., 2025)	77.5 ± 2.7	85.4 ± 0.3	86.9 ± 1.6	95.9 ± 0.6	–

ASE-Mol's values are from the GCN row of its Table 2, under Chemprop scaffold splits and its own model selection protocol (Jiang et al., 2025). MolGraph-xLSTM's values are from the AUROC columns of its MoleculeNet Table 1; it uses scaffold splits for the single-task HIV, BACE, and BBBP datasets but random splits for the multi-task Tox21 dataset, with three repetitions (Sun et al., 2025). Neither setup is identical to our fixed four-seed OGB protocol, so these rows provide task-level context only; in particular, the Tox21 scores should not be interpreted as a matched comparison.

Table 9: Per-seed results for the two additional completed OGB classification datasets (test ROC-AUC %). The mean and sample SD use the fixed seeds $\{0, 42, 100, 2026\}$.

Method	Seed	ClinTox	SIDER
GCN	0	90.78	58.75
	42	91.58	59.54
	100	88.12	59.25
	2026	91.25	59.11
	Mean ± SD	$\mathbf{90.43 \pm 1.57}$	$\mathbf{59.16 \pm 0.33}$
GMoE	0	89.62	58.98
	42	88.10	56.95
	100	89.65	54.90
	2026	91.20	55.82
	Mean ± SD	89.64 ± 1.27	56.66 ± 1.76
CoopHyperMoE	0	88.11	60.72
	42	84.51	59.08
	100	86.98	58.83
	2026	84.19	57.88
	Mean ± SD	85.95 ± 1.91	59.13 ± 1.18

Table 10: Detailed per-seed breakdown for OGB molecular regression (RMSE ↓).

Method	Run / Seed	ESOL	FreeSolv	Lipo
GCN	Seed 0	1.1735	2.6677	0.9007
	Seed 42	1.3147	2.6648	0.9113
	Seed 100	1.2332	2.7040	0.8683
	Seed 2026	1.2560	2.7264	0.9014
	Mean ± SD	$\mathbf{1.244 \pm 0.058}$	$\mathbf{2.691 \pm 0.030}$	$\mathbf{0.895 \pm 0.019}$
GMoE	Seed 0	1.3051	2.8368	0.8707
	Seed 42	1.2374	2.5403	0.9145
	Seed 100	1.2648	2.9249	0.8845
	Seed 2026	1.2455	2.3436	0.8878
	Mean ± SD	$\mathbf{1.263 \pm 0.030}$	$\mathbf{2.661 \pm 0.268}$	$\mathbf{0.889 \pm 0.018}$
CoopHyperMoE	Seed 0	1.2154	2.8103	0.9035
	Seed 42	1.3003	2.6195	0.8527
	Seed 100	1.2883	3.7375	0.8883
	Seed 2026	1.2556	2.9197	0.8879
	Mean ± SD	$\mathbf{1.265 \pm 0.038}$	$\mathbf{3.022 \pm 0.493}$	$\mathbf{0.883 \pm 0.022}$

Table 11: BACE seed-0 one-factor validation screen (ROC-AUC %). The test column documents the completed runs but was not used for model selection. This is not a four-seed comparison.

Configuration	Validation	Test
Reference	**72.31**	74.58
$\lambda_0 = 0.25$	71.03	78.77
$\lambda_0 = 0.75$	69.05	77.43
$\tau_{\text{active}} = 0.30$	71.25	78.20
$\tau_{\text{active}} = 0.70$	72.23	77.74
Refresh interval $= 50$	71.32	77.57
Refresh interval $= 1000$	70.66	79.06
$K = 2$	71.58	78.40

D RECOVERED OPTIMIZATION AND ROUTING DIAGNOSTICS

The following figures use the numerical samples stored in the chart caches of the September 10 group-meeting presentation, redrawn with consistent axes and captions. The compared models use different virtual-node settings, so these curves are not a matched ablation.

Final-checkpoint trigger audit. We instrumented the interpolation call without changing its output and evaluated the OGB validation split at the final epoch-150 checkpoints of all four seeds. Across the five MoE layers, BACE had 0/112,440 triggered node–layer evaluations and Tox21 had 0/419,100. The measured membership displacement and top-K support changes were also zero in both datasets. These are final-checkpoint, validation-only observations: they do not measure trigger activity during optimization, and they do not establish that other datasets behave identically. The audit script and per-layer counts are retained with the experiment records.

Table 12: Final-checkpoint interpolation audit on the validation splits. Counts pool four seeds and five MoE layers per dataset. Zero activity here does not measure trigger frequency during training.

Dataset	Node–layer evaluations	Triggered	Top-K changes
BACE	112,440	0	0
Tox21	419,100	0	0

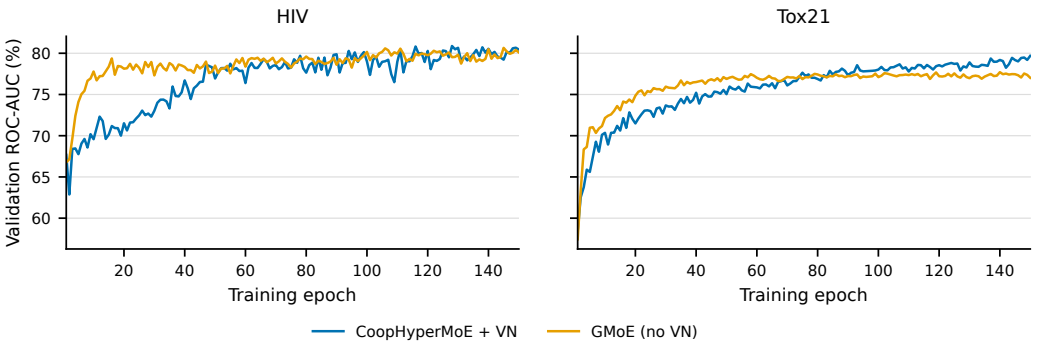

Figure 5: Mean validation ROC-AUC over 150 training epochs for four seeds. The curves come directly from per-epoch validation logs and are not smoothed; they cannot replace validation-selected test comparisons. CoopHyperMoE uses a virtual node in this archived batch, whereas the GMoE run does not.

Implementation boundaries. The expert hypergraph defines interpolation neighborhoods; it is not optimized by a separate cooperation objective. The model interpolates memberships, anneals the blend strength rather than the Gaussian bandwidth, and evaluates every expert. Cooperation

losses, interpolation of expert outputs, and conditional expert execution are possible extensions, but none is part of the evaluated model. Each would require its own implementation and matched ablation.

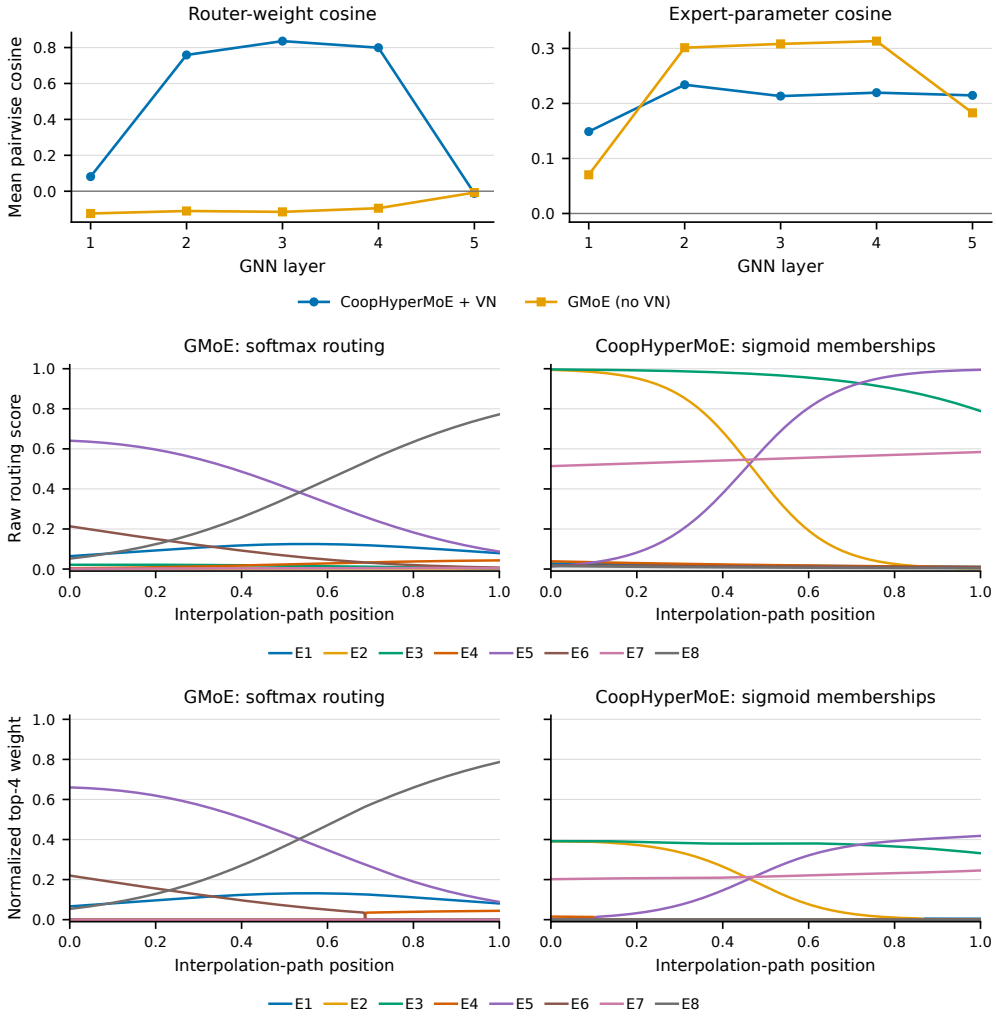

Figure 6: Routing diagnostics on Tox21. *Top:* Mean pairwise cosine similarity of router weights (left) and expert convolution parameters (right) by GNN layer at the epoch-150 checkpoint, averaged over four seeds. *Middle:* Raw routing scores along a 201-point interpolation between the hidden representations of one fixed validation-graph pair (seed 0, epoch 150, layer 1). *Bottom:* Normalized top-4 weights along the same path; GMoE and CoopHyperMoE have one and two selected-support changes, respectively, in this example. The compared models have different virtual-node settings. The path does not interpolate molecular topology, and neither this single probe nor parameter cosine similarity establishes global routing smoothness or functional expert collapse.