SPECLIFF: SPECTRAL DECOUPLING OF MOLECULAR GRAPH REPRESENTATIONS FOR ACTIVITY CLIFF PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph neural networks often fail on molecular activity cliffs, where small structural edits cause large potency changes. We trace this failure to a spectral mismatch in the standard encoder-plus-mean-pooling design. Local substituent edits produce high-frequency graph signals, while linear Laplacian propagation attenuates higher frequencies and mean pooling dilutes localized changes. Motivated by this view, we propose SPECLIFF, a dual-branch architecture that combines a low-pass scaffold branch with an explicit high-pass branch based on Chebyshev-approximated spectral graph wavelets and bridge-edge decomposition. Across all 30 MoleculeACE tasks, SPECLIFF achieves the best average RMSE and wins on the largest number of datasets compared with prior baselines. Ablation studies, frequency-response analysis, and case studies show that the high-pass branch is critical and concentrates signal around substituent edits that drive activity cliffs.

1 INTRODUCTION

Drug discovery increasingly relies on computational models to prioritize candidate molecules before costly experimental validation, from virtual screening to molecular design (Sadybekov & Katritch, 2023; Tropsha et al., 2024; Catacutan et al., 2024). In the context of bioactivity prediction, this task is often framed as supervised regression, and graph neural networks (GNNs) have become a leading approach by exploiting the graph structure of molecules (Wu et al., 2018; Wieder et al., 2020). Models are routinely trained and evaluated on independent and identically distributed splits, on which simple descriptor based models often perform comparably to or even better than deep architectures (Yang et al., 2019).

However, this aggregate performance masks a consequential failure mode: all methods, deep and simple alike, perform substantially worse on *activity cliffs*, pairs of molecules nearly identical in structure but differing in potency by an order of magnitude or more (Stumpfe et al., 2020; van Tilborg et al., 2022; Cui et al., 2026). Because activity cliffs are precisely the cases that matter most for structure activity relationship reasoning, this failure limits the practical utility of learned molecular representations in analog prioritization (Choung et al., 2023; Koscher et al., 2023), deciding which close structural variant to synthesize next during lead optimization, and selectivity engineering (Tian et al., 2025; Collie et al., 2024), where a single substituent change often separates a potent binder from a weak one against closely related targets.

Prior explanations have largely appealed to *oversquashing* and *oversmoothing*: message-passing GNNs compress or blur information as it propagates, so fine-grained local differences are lost before reaching a distant readout (Alon & Yahav, 2021; Topping et al., 2022). The oversquashing explanation has motivated graph rewiring guided by discrete curvature measures such as Ollivier-Ricci curvature, which relieve topological bottleneck edges (Topping et al., 2022). Meanwhile, oversmoothing has been addressed through normalization schemes such as PairNorm (Zhao & Akoglu, 2020a) that prevent node representations from converging.

On MoleculeACE (van Tilborg et al., 2022), we find that cliff-pair differences are structurally diverse but usually localized within small substituent-like regions. This points to a failure mode that occurs after propagation and node encoding, not during them. Unlike oversquashing, which arises from long range propagation, and oversmoothing, which arises from multilayer propagation, dilution occurs during graph level pooling: localized signals remain intact at the edited nodes but

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.

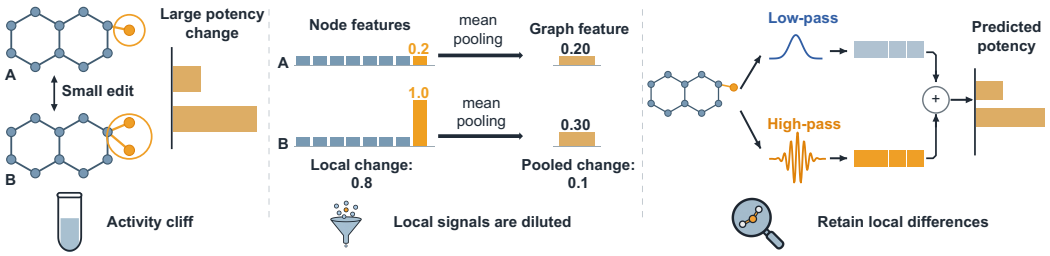

Figure 1: **Motivation of SPECLIFF.** Small structural edits can cause large potency changes (i.e., activity cliff), while mean pooling dilutes their localized signals. SPECLIFF preserves these differences through explicit low- and high-pass branches.

contribute weakly to the pooled representation. The failure instead occurs at readout: standard mean pooling compresses all node representations into a single vector, diluting a signal concentrated in a small fraction of the graph's nodes. We call this effect *dilution*, as illustrated in Figure 1. We analyze dilution using graph signal processing. Pendant-supported edits must retain energy above a Dirichlet cutoff, while linear Laplacian propagation attenuates higher frequencies and mean pooling keeps only the constant mode. A separate locality bound shows that mean pooling suppresses edits affecting few nodes when propagation depth and per-node changes remain bounded.

The dilution theory prescribes a spectral remedy: preserve the high-frequency signal that mean pooling collapses. We propose SPECLIFF, a dual-branch architecture that adds a spectral high-pass branch to a message-passing backbone. While mean pooling dilutes localized edits, the high-pass branch preserves them through wavelet filtering and magnitude pooling, guided by bridge-edge features. It predicts an additive correction to the low-pass potency estimate and is trained with a matched-pair consistency loss on structurally similar activity-cliff pairs.

Our main contributions are as follows:

- We identify pooling-induced dilution as a spectral failure mode of molecular GNNs and formalize why localized activity-cliff edits are attenuated by message passing and mean pooling (Section 3).
- We propose SPECLIFF, a dual-branch model that combines a low-pass scaffold branch with a spectral high-pass branch, trained using matched activity-cliff pairs (Section 4).
- We evaluate SPECLIFF on all 30 MoleculeACE tasks against the complete prior baseline suite, achieving the best average RMSE and the largest number of dataset-level wins (Section 5).

2 RELATED WORK

Activity cliffs and molecular property prediction. Activity cliffs are a long-studied phenomenon in medicinal chemistry: exceptions to the similarity-property principle underlying structure-activity relationships, in which minor structural changes cause disproportionately large potency shifts (Maggiora, 2006; Stumpfe et al., 2020). van Tilborg et al. (2022) introduced MoleculeACE, a curated benchmark of 30 ChEMBL (Gaulton et al., 2012) targets, and showed that graph-based, SMILES-based, and descriptor-based deep models all underperform simple descriptor-plus-classical-ML baselines on cliff compounds. A range of representation learning approaches offer potential remedies, including pretraining (Hu et al., 2020; Wang et al., 2022), knowledge-guided representations (Li et al., 2022), and oversmoothing-targeted architectures (Zhao & Akoglu, 2020b; Zhou et al., 2021; Chen et al., 2020; Gasteiger et al., 2019), which were not specifically designed for activity cliffs. Several recent methods target activity cliffs specifically: MAPCliff-WMGR (Chen et al., 2025b) combines weighted molecular graphs with sinusoidal feature mapping, MTPNet (Shu et al., 2025) incorporates multi-grained target information, and ACES-GNN (Chen et al., 2025a) supervises atom attributions using activity-cliff explanations. GraphCliff (Kim et al., 2026) specifically targets activity-cliff prediction by preserving sensitivity to small structural changes through gated short- and long-range message passing. However, these methods do not explicitly address the dilution of localized structural signals during graph-level pooling.

Oversquashing and oversmoothing. A separate line of work attributes GNN limitations on fine-grained tasks to oversquashing, the exponential compression of information across graph bottle-

necks (Alon & Yahav, 2021), and proposes curvature-guided rewiring of bottleneck edges (Topping et al., 2022). A related line of work characterizes oversmoothing as node representations becoming indistinguishable with increasing depth, attributing this to the repeated low-pass filtering effect of message passing (Oono & Suzuki, 2020). Both explanations locate the failure in the propagation stage before pooling. In contrast, we study a complementary failure mode in which localized structural edits reach the readout but are diluted by standard mean pooling.

Graph signal processing, spectral GNNs, and causal matching. Message passing has been characterized as polynomial low-pass filtering in the Laplacian spectrum (Wu et al., 2019; NT & Maehara, 2019; Defferrard et al., 2016), an interpretation that explains oversmoothing but does not by itself address how a downstream readout should treat the resulting frequency content. Spectral graph wavelets generalize wavelet analysis to irregular graphs via Chebyshev-approximated spectral kernels, avoiding explicit eigendecomposition (Hammond et al., 2011), though their use in molecular property prediction remains limited. Uncertainty principles for graph signals formalize a trade-off between vertex- and spectral-domain localization, showing that signals concentrated in the vertex domain necessarily spread across the spectral domain (Agaskar & Lu, 2013). Matched-pair and treatment-effect methods separate shared context from local effects (Shalit et al., 2017; Johansson et al., 2016) and reduce shared confounding through pairwise differencing (Imbens & Rubin, 2015), but they do not specify localized molecular effect representations.

3 A Spectral Account of the Dilution Effect

We analyze a setting relevant to activity cliffs: two molecules sharing a common scaffold but differing by a small localized subgraph. This setting is motivated by our MoleculeACE audit (see Tables 3 to 5 in Appendix): R-group attachment edges are not unusually bottleneck-like; MCS-based attribution shows that cliff-pair edits are structurally diverse; the non-common atoms are usually concentrated within small bridge-separated branches. We therefore study localized perturbations, their spectral energy, and their dilution under local propagation and mean pooling.

3.1 Preliminaries

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected, undirected molecular graph with $n = |\mathcal{V}|$ atoms. Let $\boldsymbol{A} \in \{0,1\}^{n\times n}$ be its adjacency matrix, $\boldsymbol{D} = \mathrm{diag}(d_1, \ldots, d_n)$ the degree matrix, and $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ the unnormalized graph Laplacian. We use the unnormalized Laplacian in the theoretical analysis so that the constant vector $\mathbf{1}$ is exactly the eigenvector associated with eigenvalue 0, making the correspondence between mean pooling and spectral projection in Proposition 2 exact. Because $\mathcal{G}$ is connected, $\boldsymbol{L}$ is symmetric positive semidefinite with a simple eigenvalue 0; write its eigendecomposition as $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$ with eigenvalues $0 = \lambda_1 < \lambda_2 \le \cdots \le \lambda_n$ and orthonormal eigenvectors $\boldsymbol{u}_1 = \mathbf{1}/\sqrt{n}, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n$. For a signal $\boldsymbol{x} \in \mathbb{R}^n$ defined on the vertices of $\mathcal{G}$, its graph Fourier transform is $\hat{\boldsymbol{x}} = \boldsymbol{U}^\top \boldsymbol{x}$. We refer to λ_i as the graph frequency of Fourier mode $\boldsymbol{u}_i$, and $\hat{\boldsymbol{x}}(i)$ as the corresponding Fourier coefficient.

Definition 1 (High-frequency energy fraction)**.** *For a signal $\boldsymbol{x} \in \mathbb{R}^n$ and a cutoff $\lambda_c > 0$, the high-frequency energy fraction of $\boldsymbol{x}$ above λ_c is*

$$\rho_{\lambda_c}(\boldsymbol{x}) \;=\; \frac{\sum_{i:\,\lambda_i \ge \lambda_c} \hat{\boldsymbol{x}}(i)^2}{\sum_{i=1}^{n} \hat{\boldsymbol{x}}(i)^2} \;=\; \frac{\sum_{i:\,\lambda_i \ge \lambda_c} \hat{\boldsymbol{x}}(i)^2}{\|\boldsymbol{x}\|_2^2}. \tag{1}$$

3.2 Localized perturbations require nonzero-frequency energy

We now formalize the localized-edit pattern observed in the MoleculeACE audit. Activity-cliff-driving perturbations often lie on small *pendant* subgraphs. These are branches attached to the rest of the molecule via a single bridge edge. We bound the energy above a cutoff set by the branch's Dirichlet eigenvalue. The bound depends on both the branch and the full graph spectrum; it does not imply a universal energy fraction or a size-only ordering across different branch topologies.

Definition 2 (Pendant subgraph)**.** *Let $\mathcal{S} \subset \mathcal{V}$, $|\mathcal{S}| = k$, be the vertex set of a connected subgraph attached to $\mathcal{V} \setminus \mathcal{S}$ via a single edge (u, v) with $u \in \mathcal{S}$, $v \notin \mathcal{S}$ (a bridge edge). We call $\mathcal{S}$ a pendant subgraph of size k.*

Definition 3 (Dirichlet eigenvalue of a pendant subgraph)**.** *For a pendant subgraph $\mathcal{S}$ with boundary vertex $u \in \mathcal{S}$, define the Dirichlet Laplacian $\boldsymbol{L}_{\mathcal{S}}^{D}$ as the principal submatrix of $\boldsymbol{L}$ restricted to rows and columns in $\mathcal{S}$, i.e., $\boldsymbol{L}_{\mathcal{S}}^{D} = \boldsymbol{L}[\mathcal{S}, \mathcal{S}]$. Its smallest eigenvalue*

$$\lambda_1^D(\mathcal{S}) = \min_{\boldsymbol{x} \neq 0,\, \mathrm{supp}(\boldsymbol{x}) \subseteq \mathcal{S}} \frac{\boldsymbol{x}^\top \boldsymbol{L} \boldsymbol{x}}{\|\boldsymbol{x}\|_2^2} = \lambda_{\min}(\boldsymbol{L}_{\mathcal{S}}^{D}) \tag{2}$$

is the Dirichlet eigenvalue of $\mathcal{S}$.

Lemma 1 (Dirichlet eigenvalue of a path)**.** *Let $\mathcal{S}$ be a pendant subgraph that is a simple path of k vertices $s_1, s_2, \cdots, s_k$, attached to the core at s_1 via a bridge edge (i.e., s_1 has one additional neighbor outside $\mathcal{S}$). Then*

$$\lambda_1^D(\mathcal{S}) \;=\; 2\Big(1 - \cos\Big(\frac{\pi}{2k+1}\Big)\Big). \tag{3}$$

The proof is deferred to Appendix E.1.

Proposition 1 (Spectral energy of localized perturbations)**.** *Let $\mathcal{S}$ be a pendant subgraph and $\boldsymbol{x} \neq 0$ be supported on $\mathcal{S}$. For $0 < \lambda_c < \lambda_1^D(\mathcal{S})$,*

$$\rho_{\lambda_c}(\boldsymbol{x}) \;\geq\; \frac{\lambda_1^D(\mathcal{S}) - \lambda_c}{\lambda_n - \lambda_c}. \tag{4}$$

At $\lambda_c = \lambda_1^D(\mathcal{S})/2$, the lower bound is $\lambda_1^D(\mathcal{S})/(2\lambda_n - \lambda_1^D(\mathcal{S}))$, not a universal one-half.

For path-shaped pendant branches, $\lambda_1^D(\mathcal{S}) = \Theta(1/k^2)$ (Appendix E.1), so shorter paths raise the cutoff at which nonzero energy is guaranteed. The proof and a maximum-degree bound appear in Appendix E.2.

3.3 Message passing plus mean pooling is a cascaded low-pass filter

Proposition 2 (Message passing plus mean pooling is low-pass)**.** *Under a standard Laplacian message-passing step $\boldsymbol{P} = \boldsymbol{I} - \gamma \boldsymbol{L}$ with $\gamma \in (0, 1/\lambda_n]$, the response of frequency λ_i after T linear propagation layers is scaled by $(1 - \gamma\lambda_i)^T$. This response decreases with λ_i, so higher-frequency components are attenuated more strongly with depth. Mean pooling then extracts only the zero-frequency component, making the encoder-plus-mean-pooling architecture a cascaded low-pass filter.*

The full assumptions and proof are given in Appendix E.3.

3.4 The dilution bound

Proposition 3 (Dilution of localized perturbations under standard pooling)**.** *Let $\mathcal{G}$ and $\mathcal{G}'$ share an aligned vertex set of size n. Let $\mathcal{S}$, of size k, contain all nodes with changed input features or local update rules, including endpoints of changed edges. Apply the same T-layer local message-passing encoder to both graphs. Let $\mathcal{N}_T(\mathcal{S})$ contain nodes within T hops of $\mathcal{S}$ in the union graph. Their mean-pooled representations satisfy*

$$\|\boldsymbol{z} - \boldsymbol{z}'\|_2 \;\leq\; \frac{|\mathcal{S} \cup \mathcal{N}_T(\mathcal{S})|}{n} \max_v \|\boldsymbol{h}_v(\mathcal{G}) - \boldsymbol{h}_v(\mathcal{G}')\|_2. \tag{5}$$

For fixed depth and bounded degree, the prefactor is $O(k/n)$. Vanishing sensitivity additionally requires a uniformly bounded per-node representation change. Spectral attenuation in Proposition 2 is a separate result for linear propagation on a fixed graph.

The full derivation of Proposition 3 is given in Appendix E.4.

Corollary 1 (Energy retention by a high-pass filter)**.** *Under Proposition 1, let a spectral filter h satisfy $|h(\lambda)| \geq \eta > 0$ for $\lambda \geq \lambda_1^D(\mathcal{S})/2$. Then*

$$\frac{\|h(\boldsymbol{L})\boldsymbol{x}\|_2^2}{\|\boldsymbol{x}\|_2^2} \geq \eta^2 \frac{\lambda_1^D(\mathcal{S})}{2\lambda_n - \lambda_1^D(\mathcal{S})}. \tag{6}$$

The result follows from Parseval's identity and Proposition 1. It concerns filtered node-signal energy, not guaranteed separation after a nonlinear graph readout. The bound motivates retaining high-frequency information before pooling.

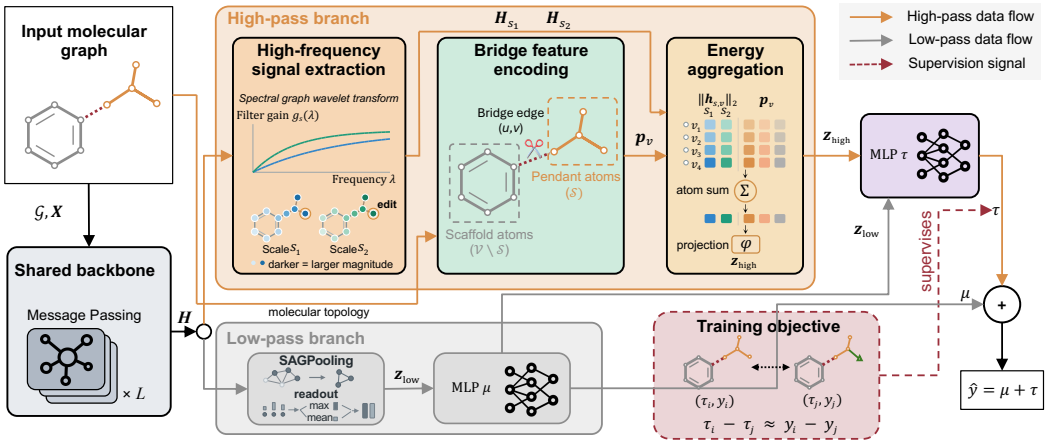

Figure 2: **The SPECLIFF architecture.** Predictions are decomposed into a low-pass scaffold term μ and an explicit high-pass substituent-effect term τ, constructed via spectral graph wavelet transform with non-vanishing high-frequency response to avoid the dilution failure. Matched-pair supervision directly aligns τ with observed activity-cliff differences.

Design implications. The analysis above motivates three design choices. First, since pendant-supported edits have nonzero energy above a Dirichlet cutoff (Proposition 1), we add an explicit channel with non-vanishing high-frequency response. Second, since mean pooling discards nonzero-frequency components (Proposition 2), this channel should not use ordinary mean pooling over signed wavelet coefficients. Third, since the dilution bound scales with the relative size of the edited region (Proposition 3), the high-frequency signal should be anchored to the local branch where the edit occurs rather than spread uniformly over the whole graph. SPECLIFF implements these choices with spectral graph wavelets, magnitude pooling, and bridge-decomposition features.

4 THE SPECLIFF METHOD

Corollary 1 motivates preserving high-frequency node-signal energy before graph-level pooling. SPECLIFF follows this motivation by decomposing prediction into a low-pass scaffold branch and an explicit high-pass substituent-effect branch, summarized in Figure 2. The scaffold branch preserves GraphCliff's readout, while the high-pass branch uses spectral graph wavelets, a bridge-decomposition prior, and energy pooling to retain localized substituent signals that mean pooling would otherwise dilute.

Formally, given a molecular graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V}$, edge set $\mathcal{E}$, and node features $\boldsymbol{X}$, SPECLIFF predicts:

$$\hat{y} = \underbrace{\mu(\boldsymbol{z}_{\text{low}})}_{\text{low-pass, Proposition 2}} + \underbrace{\tau(\boldsymbol{z}_{\text{low}}, \boldsymbol{z}_{\text{high}})}_{\text{high-pass, Corollary 1}}, \tag{7}$$

where $\mu(\cdot)$ is a scalar-valued head producing the scaffold-level potency term and $\tau(\cdot,\cdot)$ is a scalar-valued effect head producing the substituent-specific correction. For molecule i, τ_i denotes this correction term, not the full prediction. Both heads are Multi-Layer Perceptrons (MLPs), and $\boldsymbol{z}_{\text{low}}$ and $\boldsymbol{z}_{\text{high}}$ are the pooled representations from the low-pass and high-pass branches.

4.1 LOW-PASS BRANCH: SCAFFOLD-LEVEL POTENCY PREDICTION

We retain GraphCliff's encoder and SAGPooling followed by concatenated max/mean pooling to obtain $\boldsymbol{z}_{\text{low}}$; an MLP predicts μ. The mean-pooling theory motivates the added spectral branch but does not directly characterize the full inherited readout. Appendix B.1 specifies the implementation.

The inherited encoder combines edge-aware short-range updates with gated polynomial propagation. SAGPooling retains a fraction of 0.8 of the nodes and weights their embeddings before max/mean pooling. Keeping the scaffold branch unchanged preserves the backbone's molecular context while allowing the added spectral branch to provide a separate correction.

4.2 HIGH-PASS BRANCH: SUBSTITUENT-EFFECT ESTIMATION

Spectral graph wavelet transform. To construct the high-pass channel, we apply spectral graph wavelets to the node embeddings $\boldsymbol{H}$ from the final backbone layer, before SAGPooling. The implementation uses the symmetric normalized Laplacian $\boldsymbol{L}_{\text{norm}} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$, with inverse degree set to zero for isolated nodes. Write its eigenpairs as $(\boldsymbol{U}_{\text{norm}}, \boldsymbol{\Lambda}_{\text{norm}})$. At scale $s > 0$, the high-pass kernel is $g_s(\lambda) = 1 - \exp(-s\lambda)$ for $\lambda \in [0, 2]$. The transform is

$$\boldsymbol{W}_s\boldsymbol{H} = \boldsymbol{U}_{\text{norm}} g_s(\boldsymbol{\Lambda}_{\text{norm}}) \boldsymbol{U}_{\text{norm}}^{\top} \boldsymbol{H}. \tag{8}$$

Using the fixed spectral upper bound 2, a degree-K Chebyshev approximation produces $\boldsymbol{H}_{s_1}$ and $\boldsymbol{H}_{s_2}$ at two positive scales without eigendecomposition (Appendix C). The shared basis costs $O(K(|\mathcal{E}| + |\mathcal{V}|)d)$ for d-dimensional embeddings; Appendix D gives the full complexity analysis.

The two scales control how quickly the high-pass kernel rises from zero toward one across graph frequencies. Their Chebyshev coefficients are computed before training and remain fixed during optimization. Sharing the polynomial basis across scales avoids repeated sparse propagation, so the extra filtering cost grows linearly with the approximation order for a fixed graph.

Bridge features and magnitude pooling. For each node v, $\boldsymbol{p}_v$ contains a core indicator, a branch indicator, and a normalized branch identifier (Appendix B.2). To avoid signed-response cancellation, we pool the two scale magnitudes with these features:

$$\boldsymbol{e}_v = [\|\boldsymbol{h}_{s_1,v}\|_2 \,;\, \|\boldsymbol{h}_{s_2,v}\|_2 \,;\, \boldsymbol{p}_v], \qquad \boldsymbol{z}_{\text{high}} = \phi\left(\sum_{v \in \mathcal{V}} \boldsymbol{e}_v\right). \tag{9}$$

The scale responses are reduced to ℓ_2 norms, while bridge features are concatenated unchanged before summation and projection by ϕ. The effect head combines $\boldsymbol{z}_{\text{low}}$ and $\boldsymbol{z}_{\text{high}}$ to predict a scaffold-conditioned correction in Equation (7).

Both filters operate on every node, rather than only on atoms labeled as branches. Bridge features provide auxiliary structural information, not a hard mask or a learned scale-selection gate. Keeping the two scale magnitudes separate avoids cancellation between their signed responses and allows the projection to combine their graph-level contributions. Magnitude pooling does not, however, guarantee that every distinct local edit produces a distinct representation.

Conditioning the effect head on scaffold context allows the predicted correction to depend on the surrounding molecular structure. The same local substituent change need not have the same effect on different scaffolds, so the high-frequency representation is not used in isolation.

4.3 TRAINING OBJECTIVE

Preserving high-frequency energy does not guarantee that the model uses the high-pass channel for activity prediction under a standard regression loss alone. We therefore add a matched-pair consistency term that directly supervises τ. The training-pair set $\mathcal{P}$ is restricted to the training subset after validation removal. The default rule uses MoleculeACE similarities (van Tilborg et al., 2022): at least one of fingerprint, scaffold, or SMILES similarity must be ≥ 0.9, with a potency fold difference strictly greater than 10 in the implementation. No MCS-based or attachment-site filter is applied, so pairs are not restricted to content-driven edits. Appendix D.1 specifies the fallback rule and minibatch evaluation. The objective is

$$\mathcal{L} = \underbrace{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}_{\mathcal{L}_{\text{task}}} + \beta \underbrace{\frac{1}{|\mathcal{P}|}\sum_{(i,j)\in\mathcal{P}} \big((\tau_i - \tau_j) - (y_i - y_j)\big)^2}_{\mathcal{L}_{\text{pair}}}, \tag{10}$$

The pair term is defined as zero when $\mathcal{P}$ is empty. Here $\hat{y}_i = \mu_i + \tau_i$ is the model prediction for example i, y_i is the ground-truth potency, N is the number of training examples, and β is a weighting coefficient. For a structurally similar pair (i, j), much of the molecular context is shared, so the potency difference provides direct supervision for the effect difference $\tau_i - \tau_j$. The pair term therefore concentrates gradient signal on the effect head rather than supervising only the full potency prediction. Further interpretation is given in Appendix D.2.

5 EXPERIMENTS

5.1 SETUP

Datasets. We evaluate on all 30 regression datasets in MoleculeACE (van Tilborg et al., 2022), spanning ChEMBL bioactivity targets with dataset sizes ranging from 615 to 3657 molecules and cliff-compound ratios from 8.8% to 54.0%. Of the 30 datasets, 23 report inhibition constants (K_i) and 7 report half-maximal effective concentrations (EC_{50}). Both are treated as regression targets on the benchmark-provided potency scale.

Metrics. We report root mean squared error (RMSE) over all test molecules as the evaluation metric. Lower values indicate better performance.

Baselines. We compare against the complete baseline suite reported in prior activity-cliff studies (van Tilborg et al., 2022; Kim et al., 2026). This suite includes an SVM using extended-connectivity fingerprints (ECFP) as a descriptor-plus-classical-ML baseline, as well as graph-based, SMILES-based, descriptor-based neural, pretrained, and oversmoothing-mitigation models. GraphCliff (Kim et al., 2026) remains our closest architectural baseline and provides the low-pass message-passing backbone used by SPECLIFF. To ensure a fair comparison, all methods are evaluated using the same official MoleculeACE train/test splits and test RMSE metric. For model selection, we hold out 10% of the official training split as a validation set, using a deterministic stratified split based on the cliff-compound label.

Implementation and model selection. We implement SPECLIFF in PyTorch and PyTorch Geometric. For each dataset, we tune hyperparameters with Optuna using validation RMSE as the selection criterion, and evaluate the selected checkpoint once on the held-out test split. All reported SPECLIFF results use the same fixed random seed and split seed. Method-defining components, including multiscale energy pooling, bridge-decomposition features, and the MSE matched-pair loss, are kept fixed during tuning. Full training details and search ranges are provided in Appendix F, while Appendix G describes the optimization procedure and stopping criterion.

5.2 MAIN RESULTS

Table 1 reports average RMSE, rank, and #Best across the K_i, EC_{50}, and full 30-dataset splits. All baselines follow the same data splits and evaluation protocol: ACANet and MTP-Net are reproduced in-house, while the remaining results are taken from Kim et al. (2026). SPECLIFF achieves the lowest average RMSE and rank in all three groups, improving over the GraphCliff backbone from 0.665 to 0.654 overall (0.668 → 0.657 on K_i; 0.653 → 0.645 on EC_{50}). It also outperforms the recent activity-cliff methods ACANet and MTPNet, whose overall RMSE values are 0.706 and 0.723, respectively. At the dataset level, SPECLIFF yields lower RMSE than ACANet on 28 of 30 tasks and lower RMSE than MTPNet on 25 of 30 tasks. SPECLIFF is the single best

Table 1: Test RMSE on MoleculeACE, reported as the average value, average rank, and number of datasets on which each method is the single best. Lower Avg./Rank and higher #Best are better. Full per-dataset results are provided in Appendix Table 6.

	K_i ($N = 23$)			EC_{50} ($N = 7$)			All ($N = 30$)		
Method	Avg.	Rank	#Best	Avg.	Rank	#Best	Avg.	Rank	#Best
SPECLIFF	**0.657**	**2.09**	**11**	**0.645**	**1.57**	**5**	**0.654**	**1.97**	**16**
GraphCliff	0.668	3.04	3	0.653	2.36	1	0.665	2.88	4
ACANet	0.708	6.96	2	0.697	6.57	0	0.706	6.87	2
MTPNet	0.717	7.22	2	0.743	8.71	0	0.723	7.57	2
SVM	0.674	3.26	3	0.662	3.00	1	0.671	3.20	4
GINE + PairNorm	0.714	6.83	1	0.680	4.43	0	0.706	6.27	1
GINE + NodeNorm	0.724	7.15	0	0.696	7.00	0	0.718	7.12	0
GINE + Residual	0.736	9.00	0	0.701	6.71	0	0.728	8.47	0
Chemprop	0.728	8.48	0	0.737	11.07	0	0.730	9.08	0
LSTM	0.742	10.09	1	0.743	11.86	0	0.742	10.50	1
MLP	0.768	10.11	0	0.747	10.93	0	0.763	10.30	0
GINE + DropEdge	0.796	13.61	0	0.785	13.57	0	0.794	13.60	0
SCAGE (w/o 3D)	0.801	13.41	0	0.796	13.00	0	0.800	13.32	0
GCN	0.916	16.63	0	0.906	17.57	0	0.914	16.85	0
GAT	0.953	17.74	0	0.924	18.00	0	0.947	17.80	0
MPNN	1.082	18.96	0	1.135	21.14	0	1.094	19.47	0
$MolCLR_{gcn}^{pretrained}$	1.286	20.72	0	0.996	19.29	0	1.218	20.38	0
$MolCLR_{gin}^{pretrained}$	1.277	21.28	0	1.034	19.57	0	1.220	20.88	0
$ContextPred^{pretrained}$	1.622	22.96	0	1.622	23.00	0	1.622	22.97	0
KPGT	1.887	23.35	0	2.252	24.00	0	1.972	23.50	0
APPNP	0.849	15.78	0	0.821	15.57	0	0.842	15.73	0
PPNP	1.243	20.78	0	1.140	21.00	0	1.219	20.83	0
JK-Net	0.719	6.85	0	0.707	7.50	0	0.716	7.00	0
GCNII	0.808	13.72	0	0.789	12.57	0	0.803	13.45	0

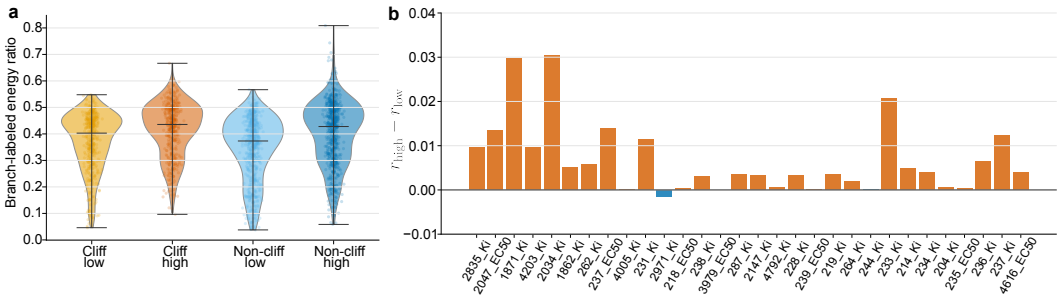

Figure 3: Frequency verification on MoleculeACE test sets. (a) Branch-labeled energy fractions for test molecules with both branch-labeled and non-branch-labeled atoms ($n = 1{,}102$). (b) Dataset-level mean shift in branch-labeled energy ratio (high-pass minus low-pass).

Table 2: Ablation study of SPECLIFF across the K_i, EC_{50}, and all 30 MoleculeACE datasets. The full per-dataset results are provided in Appendix Table 7.

	K_i ($N = 23$)		EC_{50} ($N = 7$)		All ($N = 30$)	
Method	Avg.	Rank	Avg.	Rank	Avg.	Rank
SPECLIFF (Full)	**0.657**	**1.30**	**0.645**	**1.29**	**0.654**	**1.30**
w/o high-pass branch	0.711	3.87	0.672	2.71	0.702	3.60
w/o energy pooling	0.699	3.26	0.696	4.14	0.699	3.47
w/o pair loss	0.702	3.30	0.686	3.43	0.698	3.33
fixed scale mixing	0.700	3.26	0.685	3.43	0.697	3.30

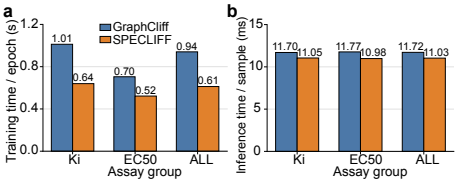

Figure 4: Efficiency comparison between GraphCliff and SPECLIFF. (a) Average training time per epoch. (b) Average inference time per sample.

method on 16 datasets, compared with 4 for GraphCliff and SVM and 2 each for ACANet and MTP-Net. The wins span both assay types ($11/23$ on K_i and $5/7$ on EC_{50}), showing that the improvement is not specific to one label type. The overall ranking separates three broad groups: GraphCliff and SVM are the strongest baselines; recent activity-cliff methods and conventional graph or sequence models form the middle group; pretrained graph encoders perform worst, with average RMSE above 1.0. The poor performance of pretrained graph encoders suggests that generic graph pretraining does not by itself address the high-frequency preservation motivated by Corollary 1. The ablations below assess the contribution of the added spectral branch.

5.3 EMPIRICAL VERIFICATION OF THE FREQUENCY RESPONSE

Corollary 1 motivates retaining high-frequency energy but does not guarantee spatial concentration after filtering. We empirically examine whether the high-pass branch assigns more energy to branch atoms than the low-pass branch by measuring, for each test molecule, the fraction of representation energy assigned to the bridge-decomposition branch atoms $\mathcal{S}$: $r_{\text{low}} = \sum_{v\in\mathcal{S}} \|\boldsymbol{h}_v^{(T)}\|_2^2 / \sum_{v\in\mathcal{V}} \|\boldsymbol{h}_v^{(T)}\|_2^2$ for the low-pass branch. For the diagnostic high-pass representation $\boldsymbol{h}_v^{\text{high}} = (\boldsymbol{h}_{s_1,v} + \boldsymbol{h}_{s_2,v})/2$, we compute r_{high} with the same formula. Both ratios use the same molecule and branch-atom set; only the representation source differs. For panel (a), a non-trivial decomposition means $0 < |\mathcal{S}|/|\mathcal{V}| < 1$, excluding molecules with no branch-labeled atoms or with all atoms branch-labeled. The subset contains $1{,}102$ test molecule records across datasets; the count is not deduplicated across targets. Panel (b) uses all test molecule records within each dataset. Figure 3 reports two views. On molecules with a non-trivial bridge decomposition, the low-pass branch assigns on average 35.6% of its energy to branch atoms, versus 40.7% for the high-pass branch, a 5.1-point increase (Figure 3a). This gap is positive for most MoleculeACE datasets when computed as $r_{\text{high}} - r_{\text{low}}$ (Figure 3b), with only a small negative outlier, indicating greater relative energy on branch-labeled atoms for most datasets.

5.4 ABLATION STUDIES

Table 2 ablates each component of SPECLIFF: (i) removes the high-pass branch entirely, predicting from $\mu(\boldsymbol{z}_{\text{low}})$ alone; (ii) replaces energy pooling with mean pooling; (iii) removes the pair loss

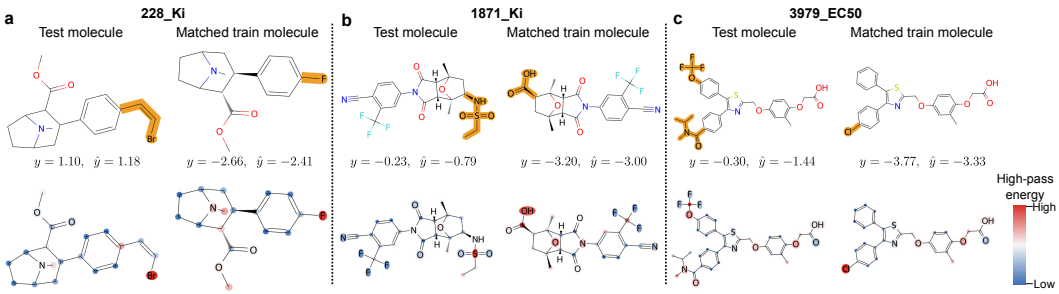

Figure 5: Case studies of activity-cliff matched pairs. Top: test molecules and matched training molecules with local structural differences highlighted. Bottom: atom-level high-pass energy from SPECLIFF, where blue indicates low energy and red indicates high energy.

($\beta = 0$); and (iv) averages the two scale responses with fixed 0.5 weights before pooling. Each ablated variant is trained from scratch using the dataset-specific hyperparameters selected for the full model, without a separate search. The no-high-pass variant retains the inherited encoder and readout and also sets $\beta = 0$, since the effect branch is absent. Checkpoints are selected by validation loss. Appendix G.1 gives the protocol.

All four ablations increase average RMSE in each dataset group. Removing the high-pass branch gives the largest increase (0.654 $\rightarrow$ 0.702 overall); the other variants yield 0.697–0.699. The results support the contribution of each component under the inherited configuration. The no-high-pass result (0.702) is not a separately tuned reproduction of the published GraphCliff result (0.665): shared architecture does not imply identical training configurations or selected checkpoints. The gap therefore cannot be assigned to architecture alone.

5.5 EFFICIENCY ANALYSIS

Under the same GPU setting, SPECLIFF has lower single-molecule inference latency than the reproduced GraphCliff implementation (11.03 vs. 11.72 ms) and lower training time per epoch (0.61 vs. 0.94 s; Figure 4). These practical timings depend on implementation and checkpoint-specific settings, including batch size, rather than providing an isolated FLOP comparison.

5.6 CASE STUDY

To complement the aggregate results, we inspect representative activity-cliff pairs from the test set. For each dataset, we select a cliff molecule in the test split and match it to a structurally similar training molecule with the same scaffold and a large activity difference. Figure 5 shows three representative cases. Although the paired molecules share similar scaffolds, their activity differences are driven by small substituent edits. The atom-level high-pass energy learned by SPECLIFF is elevated around these edited regions colored by red, indicating that the spectral branch captures local perturbation signals that are likely diluted by standard message passing and mean pooling.

6 CONCLUSION AND FUTURE WORK

We identify a structural and spectral reason why GNNs struggle with molecular activity cliffs: pendant-supported edits require nonzero-frequency energy, linear Laplacian propagation attenuates higher frequencies, and mean pooling dilutes changes confined to few nodes. This spectral view motivates SPECLIFF, which adds an explicit high-pass branch based on spectral graph wavelets and bridge-edge decomposition, together with matched-pair supervision for local substituent effects. Across MoleculeACE, SPECLIFF improves over GraphCliff and prior baselines, and our ablations and analyses show that the gain comes from preserving substituent-localized signals. Future work will extend the theoretical analysis beyond pendant-subgraph edits and separately evaluate position-driven and stereochemistry-driven cliffs. The current model can process positional variants, but the pendant-subgraph theory does not cover all such edits.

7 AI USE STATEMENT

Generative AI tools were used to improve the readability and language of the manuscript. All AI-assisted content was reviewed and verified by the authors, who take full responsibility for the final content.

8 ETHICS STATEMENT

Our study does not involve human participants, personal data, or sensitive information. The datasets and resources used are either publicly available or released under appropriate licenses. We confirm that our research does not raise any ethical concerns related to privacy, safety, fairness, or potential misuse. The contributions of this work are intended solely for advancing scientific research and are not designed or evaluated for harmful applications.

9 REPRODUCIBILITY STATEMENT

We describe the proposed method in Section 4 and the experimental setup in Section 5.1. The appendix provides the computational complexity analysis (Section D), optimization and stopping criteria (Section G), hyperparameter search settings (Section F), and per-dataset results (Section H). We include the complete implementation, processed MoleculeACE datasets, trained checkpoints, and reproduction scripts in the Supplementary Material.

REFERENCES

Ameya Agaskar and Yue M Lu. A spectral graph uncertainty principle. *IEEE Transactions on Information Theory*, 59(7):4338–4356, 2013.

Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations*, 2021.

Mitchell Black, Zhengchao Wan, Amir Nayyeri, and Yusu Wang. Understanding oversquashing in gnns through the lens of effective resistance. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 2528–2547. PMLR, 2023.

Denise B Catacutan, Jeremie Alexander, Autumn Arnold, and Jonathan M Stokes. Machine learning in preclinical drug discovery. *Nature Chemical Biology*, 20:960–973, 2024.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, pp. 1725–1735, 2020.

Xu Chen, Dazhou Yu, Liang Zhao, and Fang Liu. ACES-GNN: can graph neural network learn to explain activity cliffs? *Digital Discovery*, 4(8):2062–2074, 2025a. doi: 10.1039/D5DD00012B.

Yiwei Chen, Tingfang Wu, Yelu Jiang, Liangpeng Nie, Geng Li, Yi Zhang, Zhenglong Zhou, Jia Xu, Lijun Quan, and Qiang Lyu. MAPCliff-WMGR: Exploring activity cliffs in molecular activity prediction enhanced by weighted molecular graph representations. *Journal of Chemical Information and Modeling*, 65(23):12978–12989, 2025b. doi: 10.1021/acs.jcim.5c01915.

Oh-Hyeon Choung, Riccardo Vianello, Marwin Segler, Nikolaus Stiefl, and José Jiménez-Luna. Extracting medicinal chemistry intuition via preference machine learning. *Nature Communications*, 14(1):6651, 2023.

Gavin W Collie, Matthew A Clark, Anthony D Keefe, Andrew Madin, Jon A Read, Emma L Rivers, and Ying Zhang. Screening ultra-large encoded compound libraries leads to novel protein–ligand interactions and high selectivity. *Journal of Medicinal Chemistry*, 67(2):864, 2024.

Chao Cui, Xiaorui Su, Zaixi Zhang, Alejandro Velez-Arce, et al. Activity-cliff awareness enables robust graph learning for molecular property prediction. *Nature Communications*, 17:9376, 2026. doi: 10.1038/s41467-026-75713-2.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, 2016.

Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations*, 2019.

Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, and John P Overington. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 40(D1):D1100–D1107, 2012.

David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

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*, 2020.

Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *International Conference on Machine Learning*, pp. 3020–3029, 2016.

Hajung Kim, Jueon Park, Junseok Choe, Seungheun Baek, Hyeon Hwang, and Jaewoo Kang. Graphcliff: Short-long range gating for modeling critical activity changes caused by subtle molecular differences. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 11260–11271, 2026.

Brent A Koscher, Richard B Canty, Matthew A McDonald, Kevin P Greenman, Charles J McGill, Camille L Bilodeau, Wengong Jin, Haoyang Wu, Florence H Vermeire, Brooke Jin, et al. Autonomous, multiproperty-driven molecular discovery: From predictions to measurements and back. *Science*, 382(6677):eadi1407, 2023.

Han Li, Dan Zhao, and Jianyang Zeng. Kpgt: knowledge-guided pre-training of graph transformer for molecular property prediction. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 857–867, 2022.

Gerald M Maggiora. On outliers and activity cliffs why qsar often disappoints. *Journal of Chemical Information and Modeling*, 46(4):1535–1535, 2006.

Hoang NT and Takanori Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019.

Yann Ollivier. Ricci curvature of markov chains on metric spaces. *Journal of Functional Analysis*, 256(3):810–864, 2009.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020.

Anastasiia V Sadybekov and Vsevolod Katritch. Computational approaches streamlining drug discovery. *Nature*, 616(7958):673–685, 2023.

Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*, pp. 3076–3085, 2017.

Zishan Shu, Yufan Deng, Hongyu Zhang, Zhiwei Nie, and Jie Chen. MTPNet: Multi-grained target perception for unified activity cliff prediction. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 7733–7741, 2025. doi: 10.24963/ijcai.2025/860.

Dagmar Stumpfe, Huabin Hu, and Jürgen Bajorath. Advances in exploring activity cliffs. *Journal of Computer-Aided Molecular Design*, 34(9):929–942, 2020.

Yanan Tian, Ruiqiang Lu, Xiaoqing Gong, Wei Zhao, Yuquan Li, Xiaorui Wang, Xinming Jia, Qin Li, Yuwei Yang, Henry HY Tong, et al. Enhancing kinase-inhibitor activity and selectivity prediction through contrastive learning. *Nature Communications*, 16(1):10860, 2025.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2022.

Alexander Tropsha, Olexandr Isayev, Alexandre Varnek, Gisbert Schneider, and Artem Cherkasov. Integrating qsar modelling and deep learning in drug discovery: the emergence of deep qsar. *Nature Reviews Drug Discovery*, 23(2):141–155, 2024.

Derek van Tilborg, Alisa Alenicheva, and Francesca Grisoni. Exposing the limitations of molecular machine learning with activity cliffs. *Journal of Chemical Information and Modeling*, 62(23): 5938–5951, 2022.

Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3):279–287, 2022.

Oliver Wieder, Stefan Kohlbacher, Mélaine Kuenemann, Arthur Garon, Pierre Ducrot, Thomas Seidel, and Thierry Langer. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 37:1–12, 2020.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pp. 6861–6871. PMLR, 2019.

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(2):513–530, 2018.

Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker 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.

Lingxiao Zhao and Leman Akoglu. Pairnorm: Tackling oversmoothing in gnns. In *International Conference on Learning Representations*, 2020a. URL `https://openreview.net/forum?id=rkecl1rtwB`.

Lingxiao Zhao and Leman Akoglu. Pairnorm: Tackling oversmoothing in gnns. In *International Conference on Learning Representations*, 2020b.

Kuangqi Zhou, Yanfei Dong, Kaixin Wang, Wee Sun Lee, Bryan Hooi, Huan Xu, and Jiashi Feng. Understanding and resolving performance degradation in deep graph convolutional networks. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pp. 2728–2737, 2021.

CONTENTS

A Empirical Audit: What Actually Drives Activity Cliffs? 14

A.1 Topological bottleneck measures do not explain cliff difficulty 14

A.2 Activity cliffs are localized structural edits rather than scaffold-wide changes . . . 14

A.3 Bridge-edge decomposition recovers R-group substructure 15

B Additional Method Details 15

B.1 Scaffold encoder and readout 15

B.2 Bridge features and boundary cases 16

B.3 Dirichlet eigenvalue identity 16

B.4 Extension beyond path branches 16

C Chebyshev Implementation of SGWT 17

D Computational Complexity 17

D.1 Training-pair construction and minibatch evaluation 18

D.2 Interpretation of the pair loss 18

E Additional Proofs and Derivations 18

E.1 Proof of Lemma 1 18

E.2 Proof of Proposition 1 19

E.3 Proof of Proposition 2 19

E.4 Proof of Proposition 3 19

F Training Details 20

G Optimization and Convergence Criterion 20

G.1 Ablation protocol 21

H Per-Dataset Results 21

I Dataset and Preprocessing Details 21

Table 3: Topological bottleneck measures: R-group attachment edges vs. scaffold-internal edges. One-sided Mann-Whitney U tests evaluate the hypothesis that attachment edges are topological bottlenecks, i.e., have higher betweenness centrality or more negative Ollivier-Ricci curvature.

Measure	Edge type	n	Mean	Std	U-statistic	p-value
Betweenness centrality	Attachment (R-group)	94923	0.1278	0.0990	36057330333.5	1.0
	Scaffold-internal	1054694	0.1952	0.1437		
Ollivier-Ricci curvature	Attachment (R-group)	94923	0.0827	0.2871	66938795975.5	1.0
	Scaffold-internal	1054694	-0.1047	0.1625		

A Empirical Audit: What Actually Drives Activity Cliffs?

Before introducing our method, we establish two empirical facts about the structural origin of activity cliffs in MoleculeACE that motivate our theoretical analysis and rule out a purely topological explanation (Sections A.1 and A.2). We then introduce a purely graph theoretic decomposition (Section A.3) that formalizes the substituent sites identified by the second fact into the structural object on which our theory in Section 3 is built.

A.1 Topological bottleneck measures do not explain cliff difficulty

A natural hypothesis, consistent with the oversquashing literature (Alon & Yahav, 2021; Topping et al., 2022; Black et al., 2023), is that R-group attachment edges act as topological bottlenecks: edges lying on a disproportionate number of shortest paths that constrain information flow, or equivalently, edges with strongly negative discrete curvature. We test this hypothesis directly across the training splits of all 30 MoleculeACE datasets by computing edge betweenness centrality (Freeman, 1977), the fraction of shortest paths passing through an edge, and Ollivier-Ricci curvature (Ollivier, 2009; Topping et al., 2022), a discrete curvature measure used to identify bottleneck-like edges in graphs. We then compare attachment edges against scaffold-internal edges using one-sided Mann-Whitney U tests, with the alternative hypothesis that attachment edges have higher betweenness or more negative curvature.

As shown in Table 3, the results do not support the bottleneck hypothesis. Attachment edges have lower mean betweenness centrality than scaffold-internal edges (0.1278 vs. 0.1952), and the one-sided Mann-Whitney test gives no evidence that attachment edges have higher betweenness ($p = 1.0000$). The curvature result leads to the same conclusion: attachment edges are not more negatively curved than scaffold-internal edges, but instead have higher mean curvature (0.0827 vs. -0.1047), again with no support for the bottleneck alternative ($p = 1.0000$). Thus, both shortest-path centrality and curvature-based diagnostics fail to identify R-group attachment edges as the structural locus of cliff difficulty, suggesting that curvature-guided rewiring targets the wrong failure mode in this benchmark.

A.2 Activity cliffs are localized structural edits rather than scaffold-wide changes

We next characterize the structural nature of the difference between matched activity-cliff pairs directly, using MCS matching. For each cliff pair $(\mathcal{G}, \mathcal{G}')$, defined following van Tilborg et al. (2022) as pairs exceeding a similarity threshold with a $\geq 10\times$ potency difference, we compute the MCS between $\mathcal{G}$ and $\mathcal{G}'$ and classify the non-common atoms into two categories:

- *Content-driven*: the non-common atoms occupy the same topological attachment position in both molecules, but differ in chemical identity (e.g. a substituent replacement). We further distinguish pairs differing at a single attachment site from those differing at multiple sites.
- *Position-driven*: the non-common atoms are chemically similar or identical but occupy different topological positions on the scaffold (e.g. a positional isomer or ring-fusion change).

As shown in Table 4, content-driven changes account for a slight majority of attributed cliff pairs, with 26.7% involving a single attachment site and 25.6% involving multiple sites, while position-

Table 4: Structural attribution of activity-cliff pairs via MCS matching, aggregated over the training splits of all 30 MoleculeACE datasets. Percentages are computed over attributed pairs, except for unresolved MCS failures, which are reported over all pairs.

Category	Count	% of attributed pairs
Content-driven (single site)	23363	26.7%
Content-driven (multi-site)	22444	25.6%
Position-driven	41753	47.7%
Unresolved / MCS failure	12935	12.9%
Total	100495	100%

Table 5: Bridge-edge decomposition statistics, aggregated over MoleculeACE training splits.

Quantity	Mean	Median
Branches per molecule (B)	11.92	11.00
Branch size (atoms)	2.06	1.91
Core size (atoms)	9.49	9.00
MCS non-common atoms falling in a branch	87.19%	100%

driven changes remain substantial at 47.7%. Thus, activity cliffs are structurally diverse and cannot be reduced to a single edit type. What is common across these cases is locality: the differences are represented by non-common atoms relative to a shared MCS, and Table 5 shows that these atoms usually fall within small bridge-separated branches. The observed localization motivates the spectral analysis in Section 3, which focuses on why localized structural perturbations are diluted by standard message passing and mean pooling.

A.3 Bridge-edge decomposition recovers R-group substructure

For the structural audit, we delete all bridge edges and compute the remaining connected components. Among these components, the core maximizes overlap with the RDKit Murcko scaffold, with component size breaking ties. If scaffold extraction fails, ring atoms provide the scaffold proxy. When no scaffold atoms are available, we choose the largest component. Remaining ties follow component traversal order under the input atom indexing. All other components are labeled branches. For a connected acyclic molecule, every component is a singleton, so the first component is the core and the remaining atoms are branches. If there are no bridges, the implementation returns no branches and uses the scaffold atoms as the core, or all atoms when the scaffold is empty. Bridge finding and component traversal cost $O(|\mathcal{V}| + |\mathcal{E}|)$, excluding scaffold extraction.

The audit labels differ from the model's purely topological input features, which are defined in Appendix B.2. A component obtained by deleting all bridges can touch more than one bridge; it is not necessarily a pendant subgraph as defined in Section 3.

As shown in Table 5, the decomposition identifies multiple small branches per molecule on average (11.92 branches with mean branch size 2.06 atoms), while retaining a compact core of about 9.49 atoms. More importantly, these branches capture most of the atoms that differ between MCS-matched cliff pairs: 87.19% of MCS non-common atoms fall in a branch on average, with a median of 100%. The coverage measures overlap with audit-labeled branch atoms, not the fraction of edits confined to a single pendant subgraph. High coverage motivates examining local structural changes, but does not establish that all edits satisfy Proposition 1 or validate the model's distinct branch labels.

B Additional Method Details

B.1 Scaffold encoder and readout

We retain the GraphCliff backbone, which combines edge-aware short-range updates with gated polynomial propagation. Its T layers produce node embeddings $\boldsymbol{H} = \mathrm{Encoder}(\boldsymbol{X}, \mathcal{E})$. The polynomial propagation uses symmetric degree-normalized adjacency weights $1/\sqrt{d_u d_v}$; it does not use

a separately chosen diffusion step γ or compute graph eigenvalues. The parameter γ in Proposition 2 belongs to the simplified theoretical operator, not the implemented backbone.

The inherited readout first applies SAGPooling with a retention ratio of 0.8. Let $\mathcal{V}_{\text{pool}}$ and $\widetilde{\boldsymbol{h}}_v$ denote the retained nodes and their score-weighted embeddings. The scaffold representation is

$$\boldsymbol{z}_{\text{low}} = \left[\max_{v \in \mathcal{V}_{\text{pool}}} \widetilde{\boldsymbol{h}}_v \,;\, \frac{1}{|\mathcal{V}_{\text{pool}}|} \sum_{v \in \mathcal{V}_{\text{pool}}} \widetilde{\boldsymbol{h}}_v \right], \tag{11}$$

where the maximum is taken elementwise. An MLP produces $\mu = \text{MLP}_\mu(\boldsymbol{z}_{\text{low}})$. We retain the name low-pass branch for the inherited scaffold branch, but Propositions 2 and 3 do not establish an exact low-pass response or a dilution bound for the full SAGPooling-plus-max/mean readout. The theory motivates the added spectral channel; its benefit for the implemented backbone is evaluated empirically.

B.2 Bridge features and boundary cases

Bridge features are auxiliary inputs to the high-pass readout, not a learned nodewise scale gate. No branch-versus-core gate initialization is used, and both filters operate on every node. The implementation's energy pooling uses separate ℓ_2 magnitudes rather than squared coefficients; bridge features are neither squared nor included in those norms.

The model computes three features without scaffold extraction. After deleting all bridges, each component incident to an original bridge receives branch indicator 1 and an identifier j/B, where B is the number of such components and j follows traversal order under the input atom indexing. Other nodes have branch indicator and identifier 0. The core indicator is initialized to 1 and set to 0 for bridge-incident components of size at most $\max(2, \lfloor n/2 \rfloor)$. Core and branch indicators are therefore not mutually exclusive. Because identifiers follow input indexing, they are deterministic for cached graphs but are not canonical chemical identifiers.

For a connected acyclic graph with at least two atoms, every atom belongs to a bridge-incident singleton component, giving core indicator 0 and branch indicator 1. For a bridgeless cyclic graph or an isolated atom, all nodes retain core indicator 1, branch indicator 0, and identifier 0. The high-pass transform and magnitude pooling operate on all nodes, regardless of these indicators. Thus rigid cyclic edits are not excluded from prediction, although the bridge features cannot isolate an edit within a bridgeless ring system and the pendant-subgraph theorem does not cover such an edit. The method offers a spectral representation for those cases, not a theoretical guarantee of improved prediction.

B.3 Dirichlet eigenvalue identity

For any signal $\boldsymbol{x}$ supported on $\mathcal{S}$, we have

$$\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} = \sum_{(i,j)\in\mathcal{E}} (x_i - x_j)^2. \tag{12}$$

Since $\boldsymbol{x}$ vanishes outside $\mathcal{S}$, edges crossing from $\mathcal{S}$ to $\mathcal{V}\setminus\mathcal{S}$ contribute only through boundary terms. Therefore restricting $\boldsymbol{L}$ to the rows and columns of $\mathcal{S}$ gives exactly the Dirichlet quadratic form:

$$\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} = \boldsymbol{x}_{\mathcal{S}}^\top \boldsymbol{L}_{\mathcal{S}}^D \boldsymbol{x}_{\mathcal{S}}. \tag{13}$$

This gives

$$\min_{\boldsymbol{x}\neq 0,\, \text{supp}(\boldsymbol{x})\subseteq\mathcal{S}} \frac{\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x}}{\|\boldsymbol{x}\|_2^2} = \lambda_{\min}(\boldsymbol{L}_{\mathcal{S}}^D). \tag{14}$$

B.4 Extension beyond path branches

The Rayleigh-quotient argument in Proposition 1 applies to any pendant topology with its own Dirichlet eigenvalue $\lambda_1^D(\mathcal{S})$. The cutoff and energy bound therefore depend on topology as well as size. The path formula cannot be transferred unchanged to arbitrary trees or cyclic branches.

C Chebyshev Implementation of SGWT

We use $g_s(\lambda) = 1 - \exp(-s\lambda)$ with $s > 0$ on the normalized-Laplacian spectrum $[0, 2]$. The kernel is zero at zero frequency and increases with λ. The theoretical unnormalized-Laplacian Dirichlet thresholds are not numerical cutoffs for the normalized implementation. The same Rayleigh-quotient proof applies to a normalized operator using its own principal-submatrix eigenvalue and maximum eigenvalue.

For each kernel, we avoid explicit eigendecomposition by approximating the kernel with a degree-K Chebyshev expansion,

$$g_s(\lambda) \approx \sum_{m=0}^{K} c_{s,m} T_m(\tilde{\lambda}), \qquad \tilde{\lambda} = \lambda - 1, \tag{15}$$

where T_m is the m-th Chebyshev polynomial. With $Q = 128$ quadrature nodes $\theta_q = \pi(q+1/2)/Q$, the implementation sets

$$c_{s,m} = \frac{2 - \mathbf{1}_{m=0}}{Q} \sum_{q=0}^{Q-1} g_s(1 + \cos\theta_q) \cos(m\theta_q). \tag{16}$$

For each hyperparameter configuration, the scales and approximation order are fixed before training. The coefficients $c_{s,m}$ are computed once from the specified kernel and stored as fixed buffers; they are not learned by gradient descent. The resulting filter is a fixed-kernel Chebyshev approximation rather than a freely learned ChebNet filter.

For a spectral upper bound λ_{ub}, Chebyshev rescaling uses $2\boldsymbol{L}_{\mathrm{norm}}/\lambda_{\mathrm{ub}} - \boldsymbol{I}$. We fix $\lambda_{\mathrm{ub}} = 2$ for every graph, giving $\widetilde{\boldsymbol{L}}_{\mathrm{norm}} = \boldsymbol{L}_{\mathrm{norm}} - \boldsymbol{I} = -\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$ with spectrum in $[-1, 1]$. No eigenvalue computation or estimation is required during preprocessing or training. The transform is computed by the recurrence

$$\boldsymbol{T}_0 = \boldsymbol{H}, \qquad \boldsymbol{T}_1 = \widetilde{\boldsymbol{L}}_{\mathrm{norm}}\boldsymbol{H}, \qquad \boldsymbol{T}_m = 2\widetilde{\boldsymbol{L}}_{\mathrm{norm}}\boldsymbol{T}_{m-1} - \boldsymbol{T}_{m-2}. \tag{17}$$

The SGWT output is

$$\boldsymbol{W}_s\boldsymbol{H} \approx \sum_{m=0}^{K} c_{s,m}\boldsymbol{T}_m. \tag{18}$$

Computing this Chebyshev approximation requires only sparse matrix multiplication with the graph Laplacian and has complexity $O(K|\mathcal{E}|d)$ for d-dimensional node embeddings, comparable to a small number of message-passing layers.

D Computational Complexity

Let n and m denote the total numbers of nodes and edges in a minibatch, B the number of molecules, d the node-embedding dimension, K the Chebyshev order, and S the number of SGWT scales. In our model, $S = 2$. The two scales share one Chebyshev basis, avoiding the $O(n^3)$ cost of dense eigendecomposition. Constructing this basis requires K sparse Laplacian multiplications and costs

$$O(K(m + n)d)\,. \tag{19}$$

Combining the shared basis with the coefficients for all scales costs $O(SKnd)$. Computing the scale magnitudes and graph-level sums costs $O(Snd)$, and the high-pass projection and effect head cost $O(Bd^2)$. The total additional forward-pass cost is therefore

$$O\big(K(m + n)d + SKnd + Bd^2\big)\,. \tag{20}$$

Connected molecular graphs are sparse, so $m = \Theta(n)$. With the fixed number of scales used in our experiments, the graph-dependent part reduces to $O(Kmd)$, followed by an $O(Bd^2)$ graph-level head. The SGWT coefficients are fixed buffers rather than trainable parameters. Consequently, the added trainable parameter count is $O(d^2)$ and does not depend on molecule size, Chebyshev order, or the number of bonds. Backpropagation through the recurrence and prediction heads has the same asymptotic order as the forward computation, so training changes the constant factor but not the complexity class.

The implementation stores the shared Chebyshev basis to reuse it across scales. Its additional memory cost is $O((K+S)nd)$, which reduces to $O(Knd)$ for fixed S. Bridge features and normalized edge weights are computed once for each molecule and cached. Tarjan's bridge-finding algorithm costs $O(n+m)$ per molecule, so bridge decomposition does not add repeated work during training or inference. Matched-pair construction is also performed once before training; the pair loss is absent at inference time. These asymptotic costs complement the measured wall-clock results in Section 5.5.

D.1 Training-pair construction and minibatch evaluation

The pair cache retains a candidate only when both molecules belong to the training subset after validation removal. Default candidates satisfy at least one MoleculeACE similarity threshold ≥ 0.9 and a potency fold difference > 10. Similarity is an OR rule, not an average of the three measures. The code uses the available pEC50/pKi column for potency comparisons, falling back to the target column if necessary; pair-loss differences use the regression targets. MCS attribution in the empirical audit is separate from training-pair selection.

If the default rule yields no training pairs, the implementation checks Morgan fingerprints with radius 2 and 1024 bits. The fallback requires Tanimoto similarity ≥ 0.8 and an absolute potency-scale difference ≥ 1.0. When the dataset includes the `cliff_mol` flag, fallback candidates are restricted to training molecules marked as cliff compounds. The fallback also imposes no MCS or attachment-site constraint. If no fallback pair is available, the pair loss is zero. The fallback describes code behavior; it does not imply that every reported dataset used that rule.

During optimization, the pair term is the unweighted mean squared error over cached pairs whose two members occur in the current minibatch. A minibatch with no such pair contributes zero pair loss. No validation or test molecule contributes to the pair loss. The full-set expression in Equation (10) specifies the loss form; the implementation evaluates its minibatch counterpart rather than all cached pairs at every step.

D.2 Interpretation of the pair loss

The pair loss supervises τ through differences rather than absolute labels. For a structurally similar pair $(i,j) \in \mathcal{P}$, much of the molecular context is shared. Under the additive prediction $\hat{y} = \mu + \tau$, the corresponding scaffold terms are expected to satisfy $\mu_i \approx \mu_j$, while the label difference $y_i - y_j$ is mainly assigned to the effect difference. The constraint $\tau_i - \tau_j \approx y_i - y_j$ therefore targets the model component designed to represent local effects, instead of asking either effect term to explain a full potency value that also contains shared molecular context. Supervising effect differences rather than absolute effects is analogous to matched-pair differencing in treatment-effect estimation, where comparison removes shared factors (Imbens & Rubin, 2015).

E Additional Proofs and Derivations

E.1 Proof of Lemma 1

Proof. For $k \geq 2$, the Dirichlet Laplacian $\boldsymbol{L}_S^D$ has diagonal $(2,\dots,2,1)$ and adjacent off-diagonal entries -1. For $k=1$, it is the matrix (1). The diagonal at the attached endpoint includes the edge to the core. The eigenvalue equation can be written as

$$-x_{i-1} + 2x_i - x_{i+1} = \lambda x_i, \qquad x_0 = 0, \quad x_{k+1} = x_k. \tag{21}$$

Setting $x_i = \sin(i\theta)$ gives $\lambda = 2(1-\cos\theta)$. The terminal condition requires

$$0 = \sin((k+1)\theta) - \sin(k\theta) = 2\cos\left(\frac{(2k+1)\theta}{2}\right)\sin\left(\frac{\theta}{2}\right). \tag{22}$$

The k choices $\theta_j = (2j-1)\pi/(2k+1)$, $j = 1,\dots,k$, lie in $(0,\pi)$ and yield nonzero eigenvectors with distinct eigenvalues. They therefore give the full spectrum:

$$\lambda_j = 2\left(1 - \cos\left(\frac{(2j-1)\pi}{2k+1}\right)\right). \tag{23}$$

The smallest eigenvalue is attained at $j = 1$. Since $1 - \cos t = \Theta(t^2)$ as $t \to 0$, the formula also gives $\lambda_1^D(\mathcal{S}) = \Theta(k^{-2})$. □

E.2 Proof of Proposition 1

Proof. By Courant-Fischer, $\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} \geq \lambda_1^D(\mathcal{S})\|\boldsymbol{x}\|_2^2$ for every $\boldsymbol{x}$ supported on $\mathcal{S}$. Expanding in the Laplacian eigenbasis and splitting at λ_c,

$$\begin{aligned}\lambda_1^D(\mathcal{S})\|\boldsymbol{x}\|_2^2 &\leq \boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} \\ &= \sum_{\lambda_i < \lambda_c} \lambda_i \hat{\boldsymbol{x}}(i)^2 + \sum_{\lambda_i \geq \lambda_c} \lambda_i \hat{\boldsymbol{x}}(i)^2 \\ &\leq \lambda_c\Big(\|\boldsymbol{x}\|_2^2 - \sum_{\lambda_i \geq \lambda_c} \hat{\boldsymbol{x}}(i)^2\Big) + \lambda_n \sum_{\lambda_i \geq \lambda_c} \hat{\boldsymbol{x}}(i)^2.\end{aligned} \tag{24}$$

using $\lambda_i < \lambda_c$ on the first sum and $\lambda_i \leq \lambda_n$ on the second sum. Letting $\rho := \rho_{\lambda_c}(\boldsymbol{x}) = \sum_{\lambda_i \geq \lambda_c} \hat{\boldsymbol{x}}(i)^2 / \|\boldsymbol{x}\|_2^2$ and dividing by $\|\boldsymbol{x}\|_2^2$ gives $\lambda_1^D(\mathcal{S}) \leq \lambda_c(1-\rho) + \lambda_n \rho$, which rearranges to

$$\rho \;\geq\; \frac{\lambda_1^D(\mathcal{S}) - \lambda_c}{\lambda_n - \lambda_c}. \tag{25}$$

Since $0 < \lambda_c < \lambda_1^D(\mathcal{S}) \leq \lambda_n$, both numerator and denominator are positive. Gershgorin's theorem gives $\lambda_n \leq 2\Delta_{\max}$, where $\Delta_{\max}$ is the maximum vertex degree. Enlarging the denominator gives the valid weaker bound

$$\rho_{\lambda_c}(\boldsymbol{x}) \geq \frac{\lambda_1^D(\mathcal{S}) - \lambda_c}{2\Delta_{\max} - \lambda_c}. \tag{26}$$

Setting $\lambda_c = \lambda_1^D(\mathcal{S})/2$ yields $\rho_{\lambda_c}(\boldsymbol{x}) \geq \lambda_1^D(\mathcal{S})/(4\Delta_{\max} - \lambda_1^D(\mathcal{S}))$. Neither expression implies that half the energy lies above the cutoff. □

E.3 Proof of Proposition 2

Proof. In the linear case, $\boldsymbol{H}^{(T)} = \boldsymbol{P}^T\boldsymbol{X} = \boldsymbol{U}(\boldsymbol{I} - \gamma\boldsymbol{\Lambda})^T\boldsymbol{U}^\top\boldsymbol{X}$, so the i-th frequency component of any column of $\boldsymbol{X}$ is scaled by $(1-\gamma\lambda_i)^T$ after T layers; monotonicity in λ_i follows since $\gamma > 0$. For mean pooling, $\boldsymbol{z} = \frac{1}{n}\mathbf{1}^\top\boldsymbol{H}^{(T)} = \frac{1}{\sqrt{n}}\boldsymbol{u}_1^\top\boldsymbol{H}^{(T)}$ since $\boldsymbol{u}_1 = \mathbf{1}/\sqrt{n}$; by orthonormality of the eigenbasis, $\boldsymbol{u}_1^\top\boldsymbol{H}^{(T)}$ depends only on the $\lambda_1 = 0$ (i.e. $i = 1$) component of each column of $\boldsymbol{H}^{(T)}$, so components at $i \geq 2$ do not contribute to $\boldsymbol{z}$ at all, establishing the ideal low-pass claim. The frequencywise formula is restricted to linear propagation on a fixed graph. Nonlinearities can transfer energy between graph modes, so non-expansiveness alone does not extend the formula to arbitrary nonlinear GNNs. Mean pooling still extracts the constant component of the final embeddings. □

E.4 Proof of Proposition 3

Proof. Let $D_T = \mathcal{S} \cup \mathcal{N}_T(\mathcal{S})$. Locality implies by induction that the final embeddings agree outside D_T: a node whose T-hop receptive field contains no changed input or update rule has the same computation in both graphs. Hence

$$\begin{aligned}\|\boldsymbol{z} - \boldsymbol{z}'\|_2 &\leq \frac{1}{n}\sum_{v \in D_T} \|\boldsymbol{h}_v(\mathcal{G}) - \boldsymbol{h}_v(\mathcal{G}')\|_2 \\ &\leq \frac{|D_T|}{n}\max_v \|\boldsymbol{h}_v(\mathcal{G}) - \boldsymbol{h}_v(\mathcal{G}')\|_2.\end{aligned} \tag{27}$$

If the union graph has degree at most Δ, then $|D_T| \leq k\sum_{\ell=0}^{T}\Delta^\ell$. For fixed T and Δ, the prefactor is $O(k/n)$. The result assumes a common vertex set and local updates; molecule pairs with different atom counts require a separate alignment and pooling-normalization analysis.

For a feature perturbation $\boldsymbol{x}$ on one fixed graph, linear propagation also satisfies

$$\|\Pi_{\geq\lambda_c}\boldsymbol{P}^T\boldsymbol{x}\|_2 \leq (1-\gamma\lambda_c)^T\|\Pi_{\geq\lambda_c}\boldsymbol{x}\|_2, \tag{28}$$

where $\Pi_{\geq\lambda_c}$ projects onto frequencies at least λ_c and $0 < \gamma \leq 1/\lambda_n$. The spectral inequality concerns only the projected signal; it does not bound the full per-node change by the same factor or apply unchanged when the graph operator changes. □

Edits that change the number of atoms. For graphs with n and n' nodes, let C be an aligned set of shared atoms, R the atoms present only in $\mathcal{G}$, and A the atoms present only in $\mathcal{G}'$. Write $\boldsymbol{h}_v$ and $\boldsymbol{h}'_v$ for the final embeddings. The exact mean-pooling difference is

$$\begin{aligned} \boldsymbol{z} - \boldsymbol{z}' = \frac{1}{n}\sum_{v\in C}(\boldsymbol{h}_v - \boldsymbol{h}'_v) + \left(\frac{1}{n} - \frac{1}{n'}\right)\sum_{v\in C}\boldsymbol{h}'_v \\ + \frac{1}{n}\sum_{v\in R}\boldsymbol{h}_v - \frac{1}{n'}\sum_{v\in A}\boldsymbol{h}'_v. \end{aligned} \tag{29}$$

The second term captures the change in normalization and can remain nonzero even when the shared-atom embeddings are unchanged. The final two terms account for removed and added atoms. Consequently, Proposition 3 is stated only for aligned graphs of equal size; extending the bound to atom additions or deletions requires retaining these extra terms.

The three ingredients used above are detailed in the preceding appendix sections. Section E.1 derives the closed-form Dirichlet spectrum for path-shaped pendant branches by solving the discrete Dirichlet-Neumann eigenvalue problem. Section E.2 gives the Rayleigh-quotient argument for general pendant-supported signals and uses Gershgorin's theorem to bound the largest graph Laplacian eigenvalue by $2\Delta_{\max}$. For non-path pendant topologies, the same argument applies with the corresponding Dirichlet eigenvalue $\lambda_1^D(\mathcal{S})$; no universal ordering by branch size alone is assumed.

F Training Details

We implement SpeCliff in PyTorch and PyTorch Geometric. Unless otherwise specified, each model is trained for up to 100 epochs using AdamW with betas $(0.9, 0.95)$, cosine learning-rate decay with linear warmup, early stopping patience of 15 epochs, and minimum validation improvement threshold 10^{-4}. We use a fixed random seed of 42 and a fixed split seed of 42. For each MoleculeACE dataset, we reserve 10% of the official training split as a validation set using deterministic stratified splitting over the cliff-compound label.

Hyperparameters are tuned independently for each dataset with Optuna using a TPE sampler, minimizing validation RMSE. The searched hyperparameters include learning rate, weight decay, batch size, warmup length, minimum learning rate, number of message-passing layers, grouping parameters, dropout, SGWT scales, Chebyshev approximation order, spectral effect weight, and the pair-loss weight β. The architectural choices that define SpeCliff are not tuned: all full-model runs use the dual low-pass/high-pass architecture, multiscale energy pooling, cached bridge-decomposition features, and an unweighted MSE matched-pair consistency loss. After tuning, we load the checkpoint with the best validation loss and evaluate it once on the held-out test split.

G Optimization and Convergence Criterion

For $\beta \geq 0$, the training objective in Equation (10) is a non-negative weighted sum of two mean-squared-error terms. All operations in the high-pass branch, including the finite Chebyshev recurrence, scale-magnitude pooling, and the effect head, support end-to-end gradient optimization. We optimize all trainable parameters jointly with AdamW, use gradient clipping with maximum norm 1.0, and apply linear warmup followed by cosine learning-rate decay. The fixed SGWT coefficients and cached bridge features are not optimized.

We define practical convergence through validation performance rather than through the training loss. After every epoch, we evaluate validation MSE and save a checkpoint only when the score improves by more than 10^{-4}. Training stops after 15 consecutive epochs without such an improvement or after 100 epochs, whichever occurs first. Because RMSE is a monotone function of MSE, both metrics select the same best checkpoint. The checkpoint with the lowest validation loss is then used for the single test evaluation. Because SpeCliff is a non-convex neural network, the optimization procedure does not imply convergence to a global optimum. The fixed stopping rule is applied consistently across datasets and prevents test results from influencing optimization or model selection.

G.1 Ablation protocol

The ablation runner loads the selected full-model hyperparameters for each dataset and creates a newly initialized model for each variant; it does not load the full-model weights. Unchanged hyperparameters are inherited rather than retuned. The runner uses AdamW, warmup followed by cosine learning-rate decay, and validation-based early stopping. The training seed, split seed, validation fraction, and stopping settings are taken from the source run configuration, with defaults when a field is absent.

Removing the high-pass branch also disables the pair loss, so the variant predicts from the scaffold head alone. The other variants replace magnitude pooling with mean pooling, set the pair-loss weight to zero, or replace the separate scale responses with their equal-weight average before pooling. Each variant's best-validation checkpoint is evaluated on the test set. These experiments assess component removal under the full model's selected configuration, not the best achievable performance after independently tuning every variant.

H Per-Dataset Results

Table 6 reports the full per-dataset RMSE results on all 30 MoleculeACE tasks. SpeCliff achieves the best RMSE on 16 datasets, more than any baseline, and remains consistently competitive on both K_i and EC_{50} assays. In particular, it outperforms ACANet on 28 tasks and MTPNet on 25 tasks. These results show that the gains remain strong against recent methods designed specifically for activity cliffs, as well as GraphCliff, conventional GNNs, and pretrained graph encoders.

I Dataset and Preprocessing Details

We use the official MoleculeACE train/test splits and construct a deterministic 10% validation split from the official training set, stratified by the cliff-compound label. Molecular graphs are built from SMILES strings using RDKit, with atoms and bonds encoded following the GraphCliff preprocessing pipeline. Bridge-decomposition labels are computed once per molecule and cached as graph-level auxiliary features. Hyperparameter search details are provided in Section F.

Table 6: RMSE results across different ChEMBL target datasets. Bold and underlined values indicate the best and second-best results, respectively, based on unrounded RMSE.

Dataset	SPECLIFF	GraphCliff	SVM (ECFP)	GINE+PairNorm	GINE+NodeNorm	GINE+Residual	Chemprop	LSTM	MLP (ECFP)	GINE+DropEdge	SCAGE	GCN	GAT	MPNN	MolCLR$^{pre}_{gcn}$	MolCLR$^{pre}_{gin}$	ContextPredPre	KPGT	APPNP	PPNP	JK-Net	GCNII	ACANet	MTPNet
1862 (K_i)	0.779	0.781	0.774	0.841	0.921	0.866	0.815	**0.761**	0.878	1.006	0.875	0.942	0.987	0.948	1.068	1.051	1.351	1.668	1.008	1.463	0.972	0.960	0.827	0.770
1871 (K_i)	**0.621**	0.628	0.665	0.630	0.672	0.678	0.704	0.662	0.737	0.770	0.758	0.769	0.798	1.058	0.925	1.077	1.714	1.976	0.762	1.140	0.640	0.740	0.654	0.728
2034 (K_i)	0.729	0.747	**0.674**	0.725	0.719	0.730	0.775	0.755	0.742	0.749	0.745	0.810	0.809	0.905	1.292	1.510	1.415	1.420	0.765	1.004	0.714	0.759	0.750	0.721
2047 (EC_{50})	**0.585**	0.592	0.614	0.613	0.654	0.670	0.693	0.696	0.677	0.742	0.714	0.797	0.840	1.030	0.767	0.784	1.654	2.837	0.759	1.256	0.592	0.725	0.631	0.643
204 (K_i)	**0.684**	0.691	0.723	0.745	0.738	0.800	0.811	0.822	0.815	0.866	0.814	1.056	1.138	1.458	1.557	1.689	1.957	2.302	0.872	1.524	0.778	0.826	0.778	0.804
2147 (K_i)	**0.548**	0.560	0.576	0.693	0.779	0.711	0.649	0.646	0.723	0.785	0.910	0.840	0.966	1.024	1.905	1.226	1.266	1.675	1.107	1.977	0.666	1.047	0.611	0.690
214 (K_i)	0.635	**0.621**	0.634	0.722	0.681	0.710	0.660	0.723	0.693	0.764	0.779	1.007	1.051	1.183	1.050	1.144	1.501	1.802	0.800	1.134	0.666	0.754	0.664	0.746
218 (EC_{50})	**0.678**	0.697	0.719	0.690	0.765	0.727	0.758	0.748	0.785	0.852	0.738	0.928	0.957	1.053	1.072	1.100	1.726	2.199	0.853	1.052	0.735	0.758	0.748	0.735
219 (K_i)	**0.647**	0.664	0.710	0.720	0.702	0.747	0.692	0.780	0.756	0.833	0.876	1.026	0.979	0.903	0.984	1.022	1.661	2.121	0.886	1.071	0.740	0.816	0.704	0.756
228 (K_i)	**0.645**	0.651	0.662	0.693	0.702	0.708	0.670	0.779	0.755	0.810	0.771	0.958	1.026	1.000	1.340	1.459	1.494	1.856	0.844	1.196	0.713	0.784	0.715	0.657
231 (K_i)	0.704	0.708	0.750	0.729	0.757	0.795	0.760	0.809	1.333	0.792	0.914	0.878	0.991	1.305	1.395	1.643	1.975	2.605	0.866	1.320	0.776	0.857	**0.676**	0.808
233 (K_i)	0.784	0.786	0.774	0.796	0.791	0.794	0.799	0.850	0.845	0.861	0.824	1.056	1.066	1.074	1.229	1.299	1.811	1.670	0.904	1.297	0.777	0.837	0.834	**0.749**
234 (K_i)	0.616	0.618	0.622	0.682	0.672	0.694	0.687	0.738	0.669	0.752	0.799	0.934	0.950	0.959	1.325	1.347	1.349	1.676	0.794	1.135	0.655	0.729	**0.616**	0.718
235 (EC_{50})	0.648	0.650	**0.640**	0.647	0.656	0.688	0.708	0.727	0.718	0.712	0.694	0.901	0.869	1.058	1.028	1.313	1.615	2.670	0.761	1.067	0.672	0.693	0.669	0.690
236 (K_i)	0.708	**0.697**	0.698	0.757	0.730	0.736	0.799	0.812	0.733	0.814	0.799	0.943	1.002	1.364	1.315	1.414	1.483	2.154	0.862	1.332	0.741	0.786	0.747	0.718
237 (EC_{50})	**0.679**	0.708	0.720	0.757	0.738	0.771	0.767	0.783	0.902	0.895	0.967	1.132	1.103	1.402	1.209	1.089	1.665	1.691	0.929	1.443	0.764	1.091	0.767	1.007
237 (K_i)	0.710	0.705	**0.677**	0.738	0.728	0.753	0.743	0.773	0.722	0.782	0.762	1.112	1.085	1.053	1.338	1.345	1.564	2.158	0.829	1.340	0.747	0.769	0.755	0.705
238 (K_i)	0.604	**0.597**	0.610	0.658	0.666	0.703	0.673	0.654	0.684	0.744	0.726	0.937	0.928	1.142	1.168	1.198	1.609	2.389	0.785	1.152	0.644	0.727	0.658	0.691
239 (EC_{50})	0.679	**0.670**	0.678	0.733	0.693	0.710	0.819	0.765	0.756	0.792	0.796	0.905	0.902	1.288	0.944	1.054	1.836	2.751	0.823	1.098	0.786	0.756	0.734	0.752
244 (K_i)	**0.639**	0.668	0.715	0.786	0.758	0.765	0.726	0.800	0.795	0.847	0.892	1.075	1.088	1.659	1.897	1.849	1.694	2.126	0.990	1.604	0.758	0.857	0.720	0.749
262 (K_i)	0.734	0.752	**0.724**	0.789	0.807	0.859	0.868	0.767	0.904	0.919	0.923	0.934	0.994	1.021	1.220	1.174	1.981	2.534	0.980	1.092	0.837	0.912	0.775	0.743
264 (K_i)	**0.583**	0.619	0.615	0.643	0.658	0.663	0.637	0.665	0.672	0.744	0.705	0.855	0.896	1.082	1.067	1.081	1.661	1.514	0.794	1.077	0.658	0.704	0.608	0.652
2835 (K_i)	0.400	0.396	0.420	**0.384**	0.415	0.429	0.433	0.431	0.488	0.478	0.505	0.505	0.555	0.668	1.343	1.066	1.542	0.607	0.502	0.977	0.421	0.476	0.507	0.455
287 (K_i)	**0.666**	0.706	0.714	0.734	0.733	0.742	0.709	0.791	0.733	0.833	0.801	0.886	0.947	0.927	0.873	0.873	1.516	1.473	0.794	1.042	0.702	0.804	0.742	0.834
2971 (K_i)	**0.593**	0.615	0.605	0.683	0.647	0.692	0.745	0.689	0.674	0.679	0.745	0.781	0.803	0.973	1.818	1.232	1.764	1.337	0.840	1.384	0.654	0.860	0.687	0.596
3979 (EC_{50})	**0.619**	0.623	0.629	0.631	0.689	0.678	0.711	0.740	0.661	0.752	0.921	0.812	0.923	1.183	1.053	0.987	1.623	2.142	0.822	1.136	0.684	0.764	0.643	0.680
4005 (K_i)	**0.616**	0.617	0.646	0.667	0.674	0.650	0.709	0.764	0.680	0.704	0.705	0.875	0.861	0.998	1.045	1.121	1.557	1.506	0.721	1.079	0.622	0.806	0.662	0.708
4203 (K_i)	0.848	0.900	0.880	0.931	0.956	0.940	1.003	0.907	0.947	0.950	1.024	0.975	1.004	1.056	1.073	1.058	1.737	2.536	1.006	1.069	0.977	0.977	0.938	**0.770**
4616 (EC_{50})	**0.626**	0.634	0.635	0.692	0.675	0.660	0.704	0.739	0.727	0.748	0.740	0.867	0.873	0.934	0.899	0.909	1.238	1.474	0.797	0.929	0.714	0.738	0.690	0.692
4792 (K_i)	**0.621**	0.635	0.633	0.676	0.750	0.762	0.675	0.691	0.692	0.837	0.781	0.923	1.004	1.122	1.348	1.498	1.706	2.304	0.811	1.176	0.680	0.792	0.661	0.729

Table 7: Ablation study of SPECLIFF on all 30 MoleculeACE datasets. We report the test RMSE (↓) for the full model and four ablated variants.

Dataset	SPECLIFF (Full)	w/o high-pass	w/o energy pooling	w/o pair loss	Fixed scale mixing
1862 (K_i)	**0.779**	0.829	0.809	0.806	0.817
1871 (K_i)	**0.621**	0.830	0.772	0.729	0.692
2034 (K_i)	**0.729**	0.779	0.797	0.742	0.762
2047 (EC_{50})	**0.585**	0.660	0.681	0.685	0.700
204 (K_i)	**0.684**	0.715	0.697	0.713	0.712
2147 (K_i)	**0.548**	0.632	0.621	0.669	0.658
214 (K_i)	0.635	0.670	0.677	**0.634**	0.657
218 (EC_{50})	0.678	0.683	0.711	0.681	**0.670**
219 (K_i)	**0.647**	0.706	0.655	0.689	0.750
228 (K_i)	**0.645**	0.688	0.689	0.670	0.677
231 (K_i)	**0.704**	0.735	0.723	0.735	0.729
233 (K_i)	**0.784**	0.816	0.794	0.799	0.813
234 (K_i)	**0.616**	0.624	0.630	0.644	0.622
235 (EC_{50})	**0.648**	0.653	0.677	0.671	0.675
236 (K_i)	**0.708**	0.779	0.744	0.733	0.735
237 (EC_{50})	**0.679**	0.732	0.761	0.745	0.732
237 (K_i)	**0.710**	0.741	0.729	0.757	0.724
238 (K_i)	0.604	**0.602**	0.633	0.615	0.618
239 (EC_{50})	**0.679**	0.697	0.762	0.736	0.701
244 (K_i)	**0.639**	0.734	0.711	0.700	0.690
262 (K_i)	**0.734**	0.778	0.799	0.740	0.746
264 (K_i)	0.583	0.618	**0.576**	0.581	0.587
2835 (K_i)	**0.400**	0.487	0.558	0.502	0.522
287 (K_i)	**0.666**	0.733	0.742	0.786	0.748
2971 (K_i)	**0.593**	0.614	0.594	0.651	0.607
3979 (EC_{50})	**0.619**	0.667	0.647	0.620	0.648
4005 (K_i)	0.616	**0.605**	0.616	0.680	0.649
4203 (K_i)	**0.848**	0.947	0.878	0.912	0.975
4616 (EC_{50})	0.626	**0.614**	0.632	0.666	0.670
4792 (K_i)	0.621	0.682	0.645	0.651	**0.617**