MULTI-ANCHOR LATENT TRANSDUCTION FOR OUT-OF-DISTRIBUTION MOLECULAR PROPERTY PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting molecular properties under out-of-distribution (OOD) shift remains a major bottleneck in drug discovery. Transductive extrapolation is attractive because it predicts a test molecule relative to known training analogues, but existing molecular transduction methods such as Bilinear Transduction (BLT) have mainly been validated in hand-crafted descriptor spaces, where differences between descriptor vectors can be interpreted as property-relevant molecular transformations. This assumption does not reliably transfer to modern molecular encoders: learned embeddings are distributed, task-shaped representations whose nearest neighbors and difference vectors need not define useful analogies for property extrapolation. We introduce Multi-Anchor Latent Transduction (MALT), a molecular OOD transduction framework that makes retrieval geometry a learned component of the property predictor rather than relying on a fixed descriptor or frozen embedding space. MALT jointly trains the encoder, support geometry, and anchor-conditioned prediction rule under the same property objective, turning retrieved molecules into property-relevant latent supports rather than passive nearest neighbors. At inference time, MALT retrieves multiple training anchors in the learned latent space and predicts from their relations to the query. We instantiate two complementary prediction heads: a label-free latent-fusion head that uses only query and anchor embeddings, and a label-conditioned latent-delta head that extends BLT to learned molecular embeddings through multi-anchor updates. Across matched molecular OOD comparisons, MALT reduces average OOD error by 8.2–16.1% across different molecular representation modalities, with 14.2–30.1% reductions under label-range extrapolation, while maintaining competitive in-distribution performance.

1 INTRODUCTION

Machine learning models for molecular property prediction are increasingly used to guide drug discovery and materials design, but their value depends on generalization beyond the training distribution (Liu et al., 2021; Ji et al., 2023). In realistic discovery campaigns, models must make predictions for molecules with novel scaffolds, unfamiliar chemistries, extreme target values, or activity cliffs, where small structural changes induce large property changes (Maggiora, 2006; Van Tilborg et al., 2022). These regimes violate the standard IID assumption and often lead to sharp accuracy drops and overconfident predictions (Soleimany et al., 2021; Tossou et al., 2024).

Most molecular predictors are inductive: they learn a global structure–property map and apply it independently to each test molecule. This query-only paradigm can work in distribution, but it ignores a source of evidence central to molecular reasoning: how a new molecule relates to known analogues. Transductive prediction instead conditions each query on retrieved training examples at inference time, without updating model parameters on the test stream. This makes it attractive for OOD molecular prediction, where relative query–anchor relationships can provide evidence unavailable from the query alone (Netanyahu et al., 2023; Segal et al., 2025).

However, retrieval alone does not solve molecular transduction. Prior BLT-style methods define analogy in fixed input or descriptor-difference spaces, assuming coordinate differences encode

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.

transferable property changes. That holds for some hand-crafted descriptors but not for learned embeddings, where property-relevant directions are distributed, entangled, and shaped by pretraining or fine-tuning objectives; conversely, nearest neighbors in a pretrained space may be chemically similar yet useless as analogues for extrapolation. The result is a representation–inference mismatch: the prediction rule requires an analogical geometry that the representation was never trained to provide.

We introduce **Multi-Anchor Latent Transduction (MALT)**, a framework that adapts molecular transduction to modern encoders by jointly learning the molecular representation, retrieval geometry, and anchor-conditioned prediction operator. At inference time, MALT retrieves multiple training anchors in the learned latent space and predicts from their relation to the query. We study two complementary heads: **MALT-Fusion**, which uses query and anchor embeddings but not retrieved labels at inference, and **MALT-Delta**, which uses anchor labels through a BLT-style latent correction.

Across benchmarks covering covariate shift, target-value extrapolation, activity cliffs, and low-density quantum-property regimes, the heads expose different operating regimes: MALT-Fusion improves global extrapolation across encoder families while preserving competitive in-distribution accuracy, whereas MALT-Delta is stronger when anchor labels provide reliable local priors. Ablations with frozen retrieval spaces, fixed-neighbor label aggregation, and descriptor-space BLT baselines show the gains are not explained by retrieval alone, neighbor-label readout, or transplanting descriptor-space transduction to learned embeddings.

Contributions.

- We identify a representation–inference mismatch in BLT-style molecular transduction: descriptor-space analogies do not reliably transfer to learned molecular embeddings.
- We introduce MALT, a multi-anchor latent transduction framework that jointly learns the encoder, retrieval geometry, and anchor-conditioned prediction operator, and instantiate it with two complementary inference heads: MALT-Fusion for retrieved-label-free latent fusion and MALT-Delta for label-conditioned latent correction.
- We conduct, to our knowledge, one of the most extensive empirical studies of molecular OOD property prediction to date, spanning structural covariate shift, target-value extrapolation, low-density quantum-property extrapolation, and activity cliffs across multiple encoder families. On matched MoleculeNet comparisons, MALT reduces average OOD error by 8.2–16.1% across encoder families, with the largest improvements under target-value extrapolation, where OOD error falls by 14.2–30.1%.

2 Background and Related Work

2.1 Molecular OOD generalization

Molecular OOD shifts arise when test molecules differ from training molecules in structure, target value, assay condition, or chemical regime (Yang et al., 2019; Tossou et al., 2024). These shifts are especially consequential in molecular property prediction because deployment often targets novel scaffolds, unobserved property ranges, sparse regions of chemical-property space, or local structure–activity discontinuities (Oldenhof et al., 2022; Segal et al., 2025; Antoniuk et al., 2026; Maggiora, 2006; Van Tilborg et al., 2022).

We consider four OOD regimes with complementary failure modes. *Structural covariate shift* changes the marginal distribution over molecular structures while the task is fixed, and is instantiated by scaffold/structural splits (Yang et al., 2019; Tossou et al., 2024; van Tilborg et al., 2026). *Target-value shift* tests extrapolation to rare or extreme property values outside the training support, as in lead optimization or toxicity screening, and is instantiated by Y-splits (Soleimany et al., 2021; Segal et al., 2025). *Low-density property extrapolation* further stresses models in sparsely populated regions of continuous property space, where nearby supervision is scarce even when the underlying property remains well defined (Antoniuk et al., 2026). *Activity cliffs* test local chemical regimes in which structurally similar molecules have sharply different activities, violating the smoothness assumptions often implicit in molecular encoders and nearest-neighbor methods (Maggiora, 2006; Van Tilborg et al., 2022).

Table 1: **Design comparison of exemplar-conditioned prediction and OOD transduction methods.**

Method	OOD MPP	Multi ref.	Retrieved y/output?	Learned support geometry?	OOD transduction	Role of retrieval
General retrieval-conditioned prediction						
k-NN / kernel regression / IDW		✓	✓			Neighbor-label aggregation
Vinyals et al. (2016)		✓	✓	✓		Labelled support attention
Kim et al. (2019)		✓	✓	✓		Context (x, y) conditioning
Gorishniy et al. (2024)		✓	✓	✓		Neighbor feature-label module
Molecular retrieval-conditioned prediction						
Schimunek et al. (2023)	○	✓	✓	✓		Context-enriched support readout
Wang et al. (2025)			✓	✓		Single-reference spectral transfer
Xian et al. (2025)	○	✓	✓			Labelled prompt examples
Molecular OOD transduction						
Segal et al. (2025)	✓				✓	Single-anchor descriptor delta
MALT-Fusion (ours)	✓	✓		✓	✓	Label-free latent-anchor fusion
MALT-Delta (ours)	✓	✓	✓	✓	✓	Latent correction + anchor-label prior

✓ = yes; ○ = partial or setting-dependent.

Most molecular OOD methods improve the representation, regularization, uncertainty estimate, or deployment-time adaptation procedure, but still use a query-only prediction operator (Yang et al., 2019; Liu et al., 2025; Zhuang et al., 2023; Soleimany et al., 2021; Jiang et al., 2024; Chen et al., 2022; Bao et al., 2025; Yang et al., 2026). MALT is complementary: it changes the prediction operator itself by conditioning each query on retrieved latent supports, allowing predictions to depend on learned analogical relationships to training molecules rather than on the query representation alone.

2.2 Retrieval-conditioned prediction

Retrieval-conditioned predictors differ in how retrieved examples enter the prediction rule. Classical k-NN, kernel regression, and inverse-distance weighting aggregate neighbor labels directly. Neural set-conditioned models instead learn a context module: Vinyals et al. (2016) attend over labelled support examples, Kim et al. (2019) condition on context input–output pairs, and Gorishniy et al. (2024) uses retrieved tabular neighbors, including their labels, inside a neural retrieval module.

Molecular retrieval methods also commonly expose retrieved outputs at prediction time. Schimunek et al. (2023) uses support labels in few-shot drug discovery, Xian et al. (2025) supplies labelled molecular examples to an LLM prompt, and Wang et al. (2025) transfers spectral information from retrieved references.

Table 1 compares these methods along the five axes listed in its columns. MALT differs by combining multi-anchor transduction with learned latent retrieval: MALT-Fusion is label-free at prediction time, whereas MALT-Delta additionally uses anchor labels as a latent correction prior.

2.3 Transductive Molecular Extrapolation

We use *transductive prediction* to mean exemplar-conditioned inference at test time: each query is predicted using retrieved training examples, without updating model parameters. This follows exemplar-based transduction (Netanyahu et al., 2023; Segal et al., 2025) and differs from test-time adaptation (Chen et al., 2022; Bao et al., 2025; Yang et al., 2026), which adapts model parameters using unlabelled test data. Rather than relying only on a direct molecule–property mapping, a transductive model can compare a query molecule to training exemplars and reason analogically about how property values change with molecular differences.

BLT is the closest prior transductive method (Netanyahu et al., 2023; Segal et al., 2025). Let $\mathcal{D}^{\mathrm{tr}} = \{(x_i, y_i)\}_{i=1}^{n}$ denote the training set, where x_i is a molecular input or descriptor representation and y_i is its target property value. Instead of learning a direct predictor $h : \mathcal{X} \to \mathcal{Y}$, BLT predicts the target value of x_i from an anchor x_j and their difference vector $\Delta x = x_i - x_j$:

$$h_\theta(\Delta x, x_j) = \langle f_\theta(\Delta x),\, g_\theta(x_j) \rangle\,, \tag{1}$$

where f_θ and g_θ are neural networks. The bilinear form makes the prediction depend jointly on the molecular transformation Δx and the anchor context x_j.

BLT is trained on ordered pairs. In the extrapolative setting, these pairs are typically restricted to $y_j < y_i$, so that the anchor has a lower target value than the target:

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} \sum_{j:y_j<y_i} \ell\big(h_\theta(x_i - x_j, x_j),\, y_i\big). \tag{2}$$

This induces a constrained set of training differences,

$$\mathcal{D}^{\mathrm{tr}}_{\Delta X} = \left\{ x_i - x_j : (x_i, y_i), (x_j, y_j) \in \mathcal{D}^{\mathrm{tr}},\, y_j < y_i \right\}. \tag{3}$$

At inference time, BLT selects a single anchor whose query–anchor difference is closest to a difference observed during training:

$$x_{\mathrm{an}} = \arg \min_{x_j \in \mathcal{D}^{\mathrm{tr}}_X} \min_{\Delta x \in \mathcal{D}^{\mathrm{tr}}_{\Delta X}} \left\| x_{\mathrm{te}} - x_j - \Delta x \right\|_2. \tag{4}$$

The prediction is then

$$\hat{y}_{\mathrm{te}} = h_\theta(x_{\mathrm{te}} - x_{\mathrm{an}}, x_{\mathrm{an}}). \tag{5}$$

BLT is a principled transductive formulation, but has three limitations for molecular property prediction: it relies on fixed hand-crafted descriptors rather than learned representations; it selects a single deterministic anchor, which is brittle when several chemically reasonable anchors exist or the selected one is noisy; and its focus on extrapolative ordered pairs can cost in-distribution accuracy.

MALT addresses these by performing multi-anchor transduction in learned latent representations: it retrieves several anchors and aggregates them instead of conditioning on one. Appendices E.1 and E.2 show that the retrieved anchors are informative and diverse under distribution shift.

3 METHODS

MALT is a shared latent-anchor transduction framework: for each query it retrieves nearest training embeddings from a learned latent space and passes the query–anchor set to an anchor-conditioned head. The two heads differ only in whether retrieved anchor labels are available at inference: **MALT-Fusion** predicts from query and anchor embeddings only; **MALT-Delta** adds anchor labels through a BLT-style latent correction. Anchor labels supervise training and shape the embedding geometry in both cases. Retrieval is a non-differentiable nearest-neighbor lookup over a stop-gradient memory bank, and the encoder, retrieval geometry, and head are trained jointly with the task loss. Pseudocode, architecture, split construction, and implementation details are in Appendices A–D.

3.1 SHARED LATENT RETRIEVAL SETUP

Let

$$\mathcal{E}_\theta : m \mapsto z \in \mathbb{R}^d \tag{6}$$

denote a molecular encoder. During training, we maintain a memory bank of training-set embeddings,

$$\mathcal{B} = \left\{ (m_i, \bar{z}_i, y_i) : \bar{z}_i = \mathrm{sg}(\mathcal{E}_\theta(m_i)), (m_i, y_i) \in \mathcal{D}_{\mathrm{train}} \right\}, \tag{7}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. Stored bank embeddings therefore do not receive gradients through retrieval; gradients update the query path, encoder parameters, and prediction head.

For a query molecule m_q, we compute $z_q = \mathcal{E}_\theta(m_q)$ and retrieve the k nearest training embeddings under Euclidean distance:

$$I_q = \mathrm{TopK}_{i \in \mathcal{D}_{\mathrm{train}}} \left(-\|z_q - \bar{z}_i\|_2 \right) = \{a_1, \ldots, a_k\}. \tag{8}$$

The retrieved anchors are $\mathcal{A}_q = \{(\bar{z}_{a_j}, y_{a_j})\}_{j=1}^{k}$, with $k = 10$ by default; the query is masked from its own retrieval pool during training. The bank is refreshed once per epoch and rebuilt with the final encoder before inference. Metric variants and distance features are in the appendix.

3.2 MALT-FUSION: RETRIEVED-LABEL-FREE LATENT FUSION

MALT-Fusion uses only z_q and the retrieved anchor embeddings $\{\bar{z}_{a_j}\}_{j=1}^{k}$. For each attention head $h \in \{1, \dots, H\}$, with $d_h = d/H$,

$$\alpha_j^h = \mathrm{softmax}_j \left(\frac{(W_Q^h z_q)^\top (W_K^h \bar{z}_{a_j})}{\sqrt{d_h}} \right). \tag{9}$$

The attended anchor representation is

$$z_{\mathrm{attn}} = W_O \left[\sum_{j=1}^{k} \alpha_j^h W_V^h \bar{z}_{a_j} \right]_{h=1}^{H}, \tag{10}$$

where $[\cdot]_{h=1}^{H}$ denotes concatenation across heads. The prediction head then fuses the query and attended-anchor representations:

$$\hat{y}_q = W_{\mathrm{out}}\, \sigma(W_{\mathrm{fuse}}[z_q; z_{\mathrm{attn}}]). \tag{11}$$

For regression, $\hat{y}_q$ is real-valued and trained with MAE/L1 loss. For classification, $\hat{y}_q$ denotes logits and is trained with binary cross-entropy with logits. Because no y_{a_j} appears in Equations 9–11, MALT-Fusion cannot perform direct anchor-label aggregation. Fixed-neighbor controls testing this point are reported in Appendix E.4.

3.3 MALT-DELTA: LABEL-CONDITIONED LATENT-DELTA TRANSDUCTION

MALT-Delta is the label-conditioned counterpart to MALT-Fusion. It extends BLT-style transduction to learned molecular embeddings by retrieving multiple latent anchors, predicting anchor-specific corrections, and combining anchor-label-plus-correction estimates through learned weights. For each query–anchor pair, it forms a latent displacement

$$\Delta z_j = z_q - \bar{z}_{a_j}. \tag{12}$$

The correction input is $u_j = \Delta z_j$ for latent-only MALT-Delta, or $u_j = [\Delta z_j; e_j]$ when pairwise RDKit edit features are included. With bilinear rank r, two factorized MLPs output $f_\theta(u_j) \in \mathbb{R}^{T \times r}$ and $g_\phi(\bar{z}_{a_j}) \in \mathbb{R}^{r \times T}$. The task-wise correction is

$$\hat{\mathbf{\Delta}}_j = \mathrm{diag}\big(f_\theta(u_j) g_\phi(\bar{z}_{a_j})\big) \in \mathbb{R}^T. \tag{13}$$

Anchor corrections are combined with either learned attention weights,

$$\lambda_j = \mathrm{softmax}_j \left(\frac{(W_Q z_q)^\top (W_K \bar{z}_{a_j})}{\sqrt{d}} \right), \tag{14}$$

or a learned-temperature distance softmax,

$$\lambda_j^{(d)} = \mathrm{softmax}_j(-\|z_q - \bar{z}_{a_j}\|_2/\tau), \qquad \tau = e^s, \tag{15}$$

with $s \in \mathbb{R}$. The regression prediction is

$$\hat{y}_q = \sum_{j=1}^{k} \lambda_j \left(y_{a_j} + \hat{\mathbf{\Delta}}_j \right), \tag{16}$$

replacing λ_j by $\lambda_j^{(d)}$ for distance-softmax variants. MALT-Delta is evaluated only for regression. Weighting choices, edit-feature details, and component ablations are deferred to Appendix E.5.

3.4 TRAINING AND INFERENCE

At each epoch the training set is encoded into a stop-gradient memory bank; each minibatch is then encoded, retrieves anchors from that snapshot, and passes through either head. The task loss is backpropagated through the encoder query path and prediction head, but not through the top-k operation or the stored entries. At inference the bank is rebuilt once with the final encoder, and each test molecule is predicted from its retrieved anchors without test labels or parameter updates. Algorithms 1 and 2 (Appendix A) give both procedures.

Deployment cost. MALT adds training overhead from memory-bank refresh and anchor retrieval, while the memory bank remains compact and can be queried with either brute-force retrieval or Faiss without changing model weights. Full wall-clock, memory, bank-build, and retrieval-backend scaling results are reported in Appendix F.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Datasets. We evaluate on three regression settings that probe complementary forms of OOD generalization. **MoleculeNet** (Wu et al., 2018) includes BACE, ESOL, FreeSolv, and Lipophilicity. **BOOM** (Antoniuk et al., 2026) tests low-density label extrapolation across 8 QM9 quantum-chemical properties plus the FLASK 10K density (DEN) and heat-of-formation (HOF) tasks. **Activity Cliffs** (Van Tilborg et al., 2022) evaluates 30 ChEMBL targets where structurally similar molecules can differ by at least tenfold in potency.

Splits. MoleculeNet is evaluated under two OOD regimes. **X-split** denotes scaffold/structural OOD evaluation; construction, scaffold similarity, and overlap checks are detailed in Section C. This matters because retrieval methods can otherwise benefit from close scaffold analogues rather than structural generalization. **Y-split** follows the transductive property-extrapolation protocol of Segal et al. (2025): OOD is the top 5% of property values and ID is sampled from the remaining support. BOOM follows its low-density extrapolation protocol, and Activity Cliffs uses the benchmark's target-wise construction for local potency discontinuities.

Evaluation metrics. We report MAE on MoleculeNet and RMSE on BOOM and Activity Cliffs, averaged over 5 seeds (3 seeds for BOOM).

Baselines. We evaluate MALT with Chemprop (Heid et al., 2024), pretrained GIN (Hu et al., 2019), UniMol (Zhou et al., 2023), RDKit descriptors, and the BOOM benchmark backbones for quantum-property prediction: ChemBERTa (Chithrananda et al., 2020), ModernBERT (Warner et al., 2025), EGNN (Satorras et al., 2021), and GotenNet (Aykent & Xia, 2025). Baselines include the matched inductive heads, iMoLD (Zhuang et al., 2023), an invariant learning framework, and BLT (Segal et al., 2025) as the prior transductive baseline.

Training protocol. Each inductive encoder is first fine-tuned with its official or reference pipeline. MALT variants start from that checkpoint, keep the matched initialization and task loss, replace only the prediction head, and add retrieval from a stop-gradient training bank. Validation is restricted to the training support; no OOD validation, test labels, or test-time parameter updates are used. Full recipes are in Section D.

4.2 PERFORMANCE COMPARISON

We evaluate MALT on three benchmark families with distinct comparison sets: MoleculeNet against matched inductive encoders and BLT adaptations, BOOM against the benchmark's own baselines, and Activity Cliffs for local potency discontinuities.

MoleculeNet. Table 2 reports X-split structural OOD and Y-split label-extrapolation results. Against matched inductive encoders, MALT-Fusion improves most ID and OOD cells, cutting per-dataset OOD error by 16.1% for Chemprop, 14.4% for GIN, and 8.2% for UniMol on average. The strongest gains are on Y-splits (30.1%, 19.7%, and 14.2%, respectively). On X-splits, MALT-Fusion still yields improvements for several encoder–dataset pairs, including strong gains for GIN, but the advantage is more dataset-dependent than under label extrapolation.

BOOM. Table 3 evaluates low-density property extrapolation against BOOM benchmark baselines. Out of distribution, a MALT variant is best or tied on 8 of 10 targets: MALT-Delta-C improves over its Chemprop backbone on 8 of 10, and MALT-Fusion is best on HOMO, LUMO, and μ. In distribution, MALT-Fusion-G stays close to GotenNet, and MALT-Delta-C is best on C_v and HOF.

Table 2: **MoleculeNet regression results (MAE↓)** under covariate shift (X-split) and label extrapolation (Y-split). Entries are mean±std over 5 seeds for Test_{ID} and Test_{OOD}. Green shading marks improvement over the matched inductive encoder; orange shading marks improvement over the corresponding BLT baseline. **Bold** = best in column; underline = second best.

		BACE		ESOL		FreeSolv		Lipophilicity	
Method	**Encoder**	**Test_{ID}**	**Test_{OOD}**	**Test_{ID}**	**Test_{OOD}**	**Test_{ID}**	**Test_{OOD}**	**Test_{ID}**	**Test_{OOD}**
Covariate Shift (X-Splits)									
Inductive									
	Chemprop	0.4410 ±0.0120	0.9060 ±0.0180	0.1970 ±0.0120	0.5260 ±0.0200	0.2200 ±0.0050	0.3440 ±0.0260	0.3930 ±0.0090	0.5100 ±0.0120
	GIN	0.4970 ±0.0050	0.8060 ±0.0350	0.2790 ±0.0020	0.5670 ±0.0100	0.3430 ±0.0100	0.4360 ±0.0210	0.4610 ±0.0050	0.5700 ±0.0080
	iMoLD	0.4283 ±0.0161	0.8658 ±0.0743	0.2713 ±0.0257	0.5941 ±0.0497	0.2450 ±0.0209	0.4297 ±0.1343	0.4526 ±0.0288	0.5958 ±0.0264
	UniMol	0.3660 ±0.0210	0.5860 ±0.0380	0.1700 ±0.0070	0.4000 ±0.0090	0.1700 ±0.0050	0.2800 ±0.0100	0.3530 ±0.0070	0.4430 ±0.0120
Transductive (BLT)									
	RDKit	0.4448 ±0.0104	0.8105 ±0.0323	0.2701 ±0.0155	0.5637 ±0.0570	0.2478 ±0.0197	0.2903 ±0.0343	0.4766 ±0.0165	0.6941 ±0.0448
	GIN	0.4234 ±0.0296	0.7893 ±0.0153	0.2764 ±0.0149	0.6315 ±0.0361	0.3083 ±0.0427	0.4249 ±0.0687	0.4051 ±0.0095	0.5599 ±0.0094
	Chemprop	0.4564 ±0.0329	0.9146 ±0.0140	0.2319 ±0.0221	0.5087 ±0.0408	0.2109 ±0.0154	0.3238 ±0.0347	0.3781 ±0.0054	0.5170 ±0.0223
	UniMol	0.3940 ±0.0225	0.6316 ±0.0422	0.2052 ±0.0157	0.4461 ±0.0114	0.1563 ±0.0054	**0.2343** ±0.0236	0.3717 ±0.0065	0.4817 ±0.0171
Transductive (Ours — MALT-Fusion)									
	RDKit	0.3306 ±0.0236	0.6079 ±0.0530	0.2188 ±0.0120	**0.3658** ±0.0120	**0.1266** ±0.0081	0.2391 ±0.0163	0.3879 ±0.0091	0.6138 ±0.0091
	Chemprop	**0.2920** ±0.0070	0.7370 ±0.0480	0.1850 ±0.0030	0.4670 ±0.0160	0.1690 ±0.0050	0.4160 ±0.0730	0.3290 ±0.0070	0.5140 ±0.0210
	GIN	0.3060 ±0.0140	0.6450 ±0.0340	0.2210 ±0.0120	0.5820 ±0.0070	0.2010 ±0.0120	0.3760 ±0.0230	0.3480 ±0.0050	0.5400 ±0.0150
	UniMol	0.3342 ±0.0100	0.5517 ±0.0195	**0.1682** ±0.0113	0.4055 ±0.0113	0.1703 ±0.0081	0.2720 ±0.0240	0.3535 ±0.0101	**0.4350** ±0.0170
Transductive (Ours — MALT-Delta)									
	Chemprop	0.3740 ±0.0130	0.6320 ±0.0950	0.2060 ±0.0100	0.4760 ±0.0070	0.1740 ±0.0100	0.2580 ±0.0430	0.3260 ±0.0100	0.5180 ±0.0290
	GIN	0.3530 ±0.0230	**0.5080** ±0.0190	0.1950 ±0.0030	0.4150 ±0.0320	0.1750 ±0.0130	0.3530 ±0.0170	**0.3160** ±0.0120	0.4640 ±0.0050
	UniMol	0.3399 ±0.0068	0.5595 ±0.0069	0.1976 ±0.0066	0.3831 ±0.0122	0.1368 ±0.0159	0.2547 ±0.0080	0.3601 ±0.0185	0.4705 ±0.0177
Label Shift (Y-Splits)									
Inductive									
	Chemprop	0.4200 ±0.0060	1.1480 ±0.0290	0.1990 ±0.0070	0.4480 ±0.0170	0.1980 ±0.0120	0.4000 ±0.0320	0.3820 ±0.0070	0.6890 ±0.0380
	GIN	0.5080 ±0.0070	0.8020 ±0.0120	0.2430 ±0.0030	0.5390 ±0.0160	0.3480 ±0.0100	0.4880 ±0.0440	0.4210 ±0.0050	0.7620 ±0.0260
	iMoLD	0.3786 ±0.0289	0.9289 ±0.0269	0.2382 ±0.0286	0.5429 ±0.1127	0.1964 ±0.0071	0.4401 ±0.0524	0.3787 ±0.0107	0.7957 ±0.0482
	UniMol	0.3630 ±0.0150	0.9290 ±0.0520	0.1790 ±0.0060	0.5190 ±0.0170	0.1560 ±0.0110	0.2010 ±0.0270	0.3480 ±0.0060	0.7220 ±0.0130
Transductive (BLT)									
	RDKit	0.4956 ±0.0202	0.7576 ±0.0471	0.2356 ±0.0102	0.5252 ±0.0143	0.3787 ±0.0368	0.4710 ±0.0380	0.4367 ±0.0183	0.7654 ±0.0255
	GIN	0.4674 ±0.0253	0.7287 ±0.0274	0.2175 ±0.0135	0.6180 ±0.0193	0.3440 ±0.0489	0.4202 ±0.0217	0.3855 ±0.0054	0.6106 ±0.0099
	Chemprop	0.4728 ±0.0287	0.7860 ±0.0365	0.1926 ±0.0106	0.4346 ±0.0199	0.2561 ±0.0205	0.3501 ±0.0728	0.3580 ±0.0106	0.6024 ±0.0196
	UniMol	0.4084 ±0.0397	0.9289 ±0.0329	0.1920 ±0.0051	0.5450 ±0.0424	0.1788 ±0.0071	0.2059 ±0.0309	0.3614 ±0.0037	0.7549 ±0.0220
Transductive (Ours — MALT-Fusion)									
	RDKit	0.3819 ±0.0254	0.7833 ±0.0285	0.1862 ±0.0077	0.4906 ±0.0172	0.1492 ±0.0080	0.3305 ±0.0503	0.3246 ±0.0103	0.7430 ±0.0103
	Chemprop	**0.3520** ±0.0070	0.8000 ±0.0390	**0.1570** ±0.0020	0.3610 ±0.0170	0.1970 ±0.0070	0.2200 ±0.0220	**0.3120** ±0.0060	0.5120 ±0.0250
	GIN	0.3610 ±0.0110	0.7410 ±0.0360	0.1790 ±0.0090	0.5180 ±0.0290	0.1930 ±0.0200	0.3360 ±0.0170	0.3340 ±0.0060	**0.4870** ±0.0070
	UniMol	0.3548 ±0.0223	0.8054 ±0.0293	0.1823 ±0.0053	0.4941 ±0.0298	0.1588 ±0.0140	**0.1466** ±0.0159	0.3447 ±0.0091	0.6394 ±0.0171
Transductive (Ours — MALT-Delta)									
	Chemprop	0.3760 ±0.0150	0.7210 ±0.0430	0.1690 ±0.0050	**0.3480** ±0.0210	0.1540 ±0.0070	0.2050 ±0.0390	0.3240 ±0.0090	0.5250 ±0.0150
	GIN	0.4140 ±0.0270	**0.6770** ±0.0410	0.1800 ±0.0070	0.4800 ±0.0220	0.1600 ±0.0080	0.2400 ±0.0210	0.3300 ±0.0110	0.5200 ±0.0230
	UniMol	0.3671 ±0.0220	0.8970 ±0.0613	0.1699 ±0.0070	0.5235 ±0.0246	**0.1254** ±0.0168	0.2622 ±0.0247	0.3610 ±0.0304	0.6995 ±0.0324

Table 3: **BOOM benchmark results (RMSE↓).** 3 seed average ID and OOD results are reported side by side. **Bold** = best in column; underline = second best.

	In-distribution (ID)										Out-of-distribution (OOD)									
	QM9								FLASK 10K		QM9								FLASK 10K	
Model	HOMO	LUMO	GAP	μ	α	$\langle R^2 \rangle$	C_v	ZPVE	DEN	HOF	HOMO	LUMO	GAP	μ	α	$\langle R^2 \rangle$	C_v	ZPVE	DEN	HOF
RandomForest	.0060	.0073	.0084	0.67	0.97	52.6	0.42	.0011	.024	24.2	.0305	.0371	.0373	2.90	8.45	363	3.35	.0230	.180	140
MLP	.0076	.0109	.0138	4.30	1.29	66.7	12.3	.0017	.031	27.6	.0207	.0246	.0409	2.79	13.9	365	1.50	.0036	.098	46.5
ChemBERTa	.0072	.0102	.0120	0.72	1.25	55.5	0.63	.0041	.019	28.1	.0237	.0251	.0306	2.65	5.97	292	2.11	.0188	.113	108
ModernBERT	.0077	.0100	.0120	0.74	1.21	59.3	0.51	.0009	.017	28.5	.0212	.0260	.0353	2.76	2.77	210	**0.780**	**.0025**	.061	118
Chemprop	.0041	.0048	.0060	0.55	0.65	32.9	0.25	.0018	.009	13.9	.0189	.0185	.0251	2.34	4.94	263	2.34	.0063	.067	92.3
EGNN	.0044	.0045	.0062	**0.47**	0.57	**18.9**	0.29	**.0006**	.009	14.2	.0202	.0213	.0295	2.44	4.95	**160**	2.06	.0048	.059	36.6
GotenNet	**.0030**	**.0042**	**.0054**	0.48	**0.43**	29.9	0.175	.0010	**.005**	7.12	.0197	.0198	.0265	2.49	3.52	240	1.00	.0038	.013	48.8
Fusion-C	.0043	.0046	.0064	0.55	0.68	37.2	0.246	.0013	.008	14.5	**.0174**	.0176	.0258	**2.21**	6.14	279	2.79	.0068	.036	97.6
Fusion-G	.0032	.0044	.0055	0.48	0.46	29.6	0.178	.0011	**.005**	7.43	.0189	**.0173**	.0271	2.43	3.53	252	1.01	.0045	.013	48.4
Delta-C	.0040	.0045	.0058	0.58	**0.43**	31.7	**0.170**	.0009	.006	**6.13**	.0189	.0186	**.0240**	2.30	**1.86**	187	0.788	**.0025**	.035	**21.6**
Delta-G	.0034	.0046	.0056	0.48	**0.43**	29.8	0.176	.0011	**.005**	7.49	.0181	.0180	.0272	2.30	3.32	238	0.959	.0044	**.012**	51.3

Fusion = MALT-Fusion; *Delta* = MALT-Delta; *C* = Chemprop; *G* = GotenNet. RandomForest, MLP, ChemBERTa, ModernBERT and EGNN results are reported from Antoniuk et al. (2026).

Activity Cliffs. Table 4 tests local structure–activity discontinuities. Gains are less uniform than on MoleculeNet but remain positive for GIN and Chemprop. MALT-Delta gives the larger GIN gain and the most UniMol top-2 OOD rankings, suggesting that anchor-label priors help in local discontinuity regimes. Full per-target results are provided in Section E.6.

Table 4: Activity Cliffs (30 ChEMBL targets, 5 seeds). RMSE $\Delta\%$ is the median over targets relative to the matched inductive baseline (negative = improvement). *Top-2 OOD* is the number of targets where the method ranks ≤ 2 in OOD RMSE within the 9-method field (3 base + 3 Fusion + 3 Delta). **Bold** marks the better of {MALT-Fusion, MALT-Delta} per cell. Both heads improve over the baseline on GIN/Chemprop; on UniMol the base is already strong, but MALT-Delta-UniMol still captures the most top-2 slots (23/30).

	ID RMSE $\Delta\%$		OOD RMSE $\Delta\%$		Top-2 OOD (of 30)		
Encoder	**Fusion**	**Delta**	**Fusion**	**Delta**	**Base**	**Fusion**	**Delta**
GIN	-13.4	$\mathbf{-17.0}$	-14.9	$\mathbf{-15.8}$	0	4	**5**
Chemprop	$\mathbf{-9.7}$	-8.1	$\mathbf{-8.6}$	-6.1	0	**4**	3
UniMol	$\mathbf{-2.1}$	$\mathbf{-2.1}$	-0.9	$\mathbf{-1.6}$	5	16	**23**

4.3 Robustness Analysis

We analyze MALT across harder OOD regimes by stratifying MoleculeNet test molecules along two difficulty axes: structural novelty tertiles **T1–T3**, measured by distance to the nearest training scaffold, and OOD target-value extremity quartiles **Q1–Q4**; definitions are provided in Section E.10. Figures 1 and 2 show that inductive baselines degrade more sharply as queries move farther from training support or deeper into the OOD tail, while MALT remains flatter. The largest gains occur in the hardest label-extrapolation strata, supporting the interpretation that multi-anchor fusion is most useful when query-only encoders regress toward the training distribution. Retrieval quality matters: holding the trained MALT-Fusion-GIN model fixed and changing only its ten test-time anchors, Y-split OOD MAE rises from nearest to ranks 50–59 ($+15\%$), random ($+39\%$), and farthest anchors ($+135\%$), so MALT relies on what it retrieves rather than acting as a query-only predictor (Section E.11).

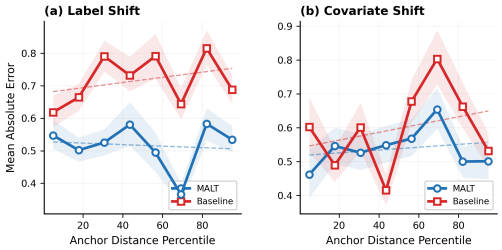


Figure 1: **Anchor-distance robustness.** MAE as a function of anchor-distance percentile.

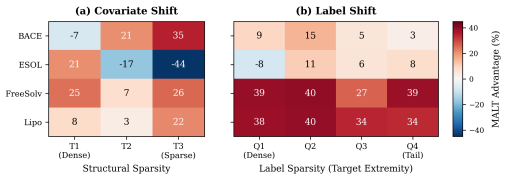


Figure 2: **OOD severity robustness.** Relative advantage of MALT-Fusion-GIN under structural and label severity strata.

4.4 Ablation Studies

The ablations test whether MALT's gains come from the proposed representation–retrieval–fusion mechanism rather than simpler retrieval alternatives. Table 5 shows that jointly adapting the retrieval space lowers mean OOD MAE relative to frozen retrieval, indicating that MALT benefits from learning a property-aware geometry. A direct probe of the representation–inference mismatch agrees: on identical held-out molecule pairs, a linear map from displacement Δz to property change Δy leaves a relative residual of 0.083 in the MALT-Fusion-GIN space, versus 0.133 for the fine-tuned GIN encoder and 0.291 for BLT's RDKit descriptors (Table 20). Table 6 then fixes the retrieved neighbors and varies only the prediction rule: direct label aggregation methods, including k-NN, inverse-distance weighting, kernel regression, 1-NN, and attentive label readout, remain worse than latent fusion. Thus, the gain comes from fusing query and anchor representations before prediction rather than directly reading out nearby labels.

Additional appendix ablations cover full encoder-training sweeps (Section E.3.2), anchor-count selection (Table 15), fixed-neighbor label aggregation (Section E.4), anchor-selection rules (Section E.3.1),

and visual diagnostics of error diversity, parity, and post-training neighborhood contraction (Sections E.7 to E.9).

Table 5: **Joint training versus frozen retrieval.** Mean OOD MAE on MoleculeNet Y-split; lower is better.

Encoder	Frozen	Joint	Δ
Chemprop	0.591	**0.473**	−20%
GIN	0.588	**0.520**	−12%
UniMol	1.128	**0.473**	−58%

Table 6: **Latent fusion versus label aggregation.** Mean OOD MAE under fixed retrieved neighbors; lower is better. Pretrained GIN encoder, jointly fine-tuned with each head; the label-readout rows share one retrieved k=10 neighbor set. Values are means over 4 MoleculeNet datasets × 5 seeds (Y-split); per-dataset values in Section E.4.

Prediction rule	OOD MAE ↓	Δ vs MALT
k-NN average	0.753	+44.8%
Scaled dot-product (Vaswani et al., 2017)	0.745	+43.3%
Nadaraya–Watson	0.740	+42.3%
IDW	0.723	+39.0%
1-NN	0.605	+16.3%
Attentive LR (Kim et al., 2019)	0.603	+16.0%
MALT latent fusion	**0.520**	0%

4.5 MALT-Delta operating regime

We next ask when prediction-time anchor labels help. MALT-Delta uses retrieved labels through a learned latent correction, while MALT-Fusion uses only query–anchor embeddings, so their comparison isolates the value of explicit anchor-label priors. Because Equations 9–11 contain no y_{a_j}, MALT-Fusion cannot directly aggregate anchor labels. For GIN and Chemprop, the raw top-k label mean over the same anchors is substantially worse than both learned heads, yet MALT-Delta beats MALT-Fusion in $33/40$ and $25/40$ MoleculeNet OOD seed-cells, respectively. In contrast, UniMol shows the failure mode: MALT-Delta wins only $10/40$ seed-cells and has worse aggregate OOD MAE (0.506 vs. 0.469). Thus, anchor labels help mainly when they provide reliable local priors; MALT-Fusion is the safer default when pretrained geometry is strong or global label extrapolation dominates.

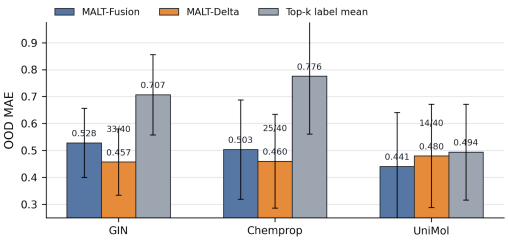


Figure 3: **Anchor-label evidence.** Aggregate MoleculeNet OOD MAE across 8 cells × 5 seeds. MALT-Delta helps for GIN and Chemprop, but not UniMol. The raw top-k label mean uses the same MALT retrieval set; numbers above Delta bars show MALT-Delta wins over MALT-Fusion. UniMol bars: earlier runs without the fine-tuned initialization (matched-init results in the text).

5 Conclusion

We introduced MALT, a multi-anchor transductive framework that enhances molecular property prediction in out-of-distribution settings by performing analogical reasoning in learned latent spaces. Our model-agnostic approach operates as a plug-and-play module for any encoder, outperforming inductive baselines on most OOD benchmarks and prior single-anchor transductive methods.

Limitations. While MALT improves OOD molecular property prediction across several benchmarks, MALT adds memory-bank storage and nearest-neighbor retrieval overhead relative to query-only predictors, which may require approximate retrieval or additional engineering for very large molecular libraries. Scientifically, MALT depends on the training bank holding property-relevant analogues: gains shrink when the pretrained encoder is already strong and when retrieved neighbors are structurally similar but carry contradictory labels, a failure mode to which MALT-Delta is more exposed because it conditions on anchor labels directly. MALT also provides no calibrated OOD uncertainty, abstention, or automatic query-only fallback, so deciding when retrieved evidence should be trusted remains open.

ETHICS STATEMENT

This work advances out-of-distribution molecular property prediction with direct applications in drug discovery. Reliable ADMET and potency predictions accelerate lead-optimisation triage, enabling computational pre-screening before committing to synthesis. By addressing activity cliffs—regimes where small structural perturbations cause large, abrupt changes in measured activity—the method can surface high-value candidates that nearest-neighbor heuristics routinely miss, thereby reducing unnecessary wet-lab burden during early-stage campaigns.

Dual-use and biosecurity. The same predictive capability that expedites benign drug discovery could in principle assist the design of chemical or biological weapons by screening candidate compounds for harmful potency profiles. We take this risk seriously. Our mitigation strategy is threefold: (i) we train and evaluate exclusively on public, open-access benchmarks (BOOM, MoleculeNet, ChEMBL); (ii) no synthesis pathways or reaction conditions are disclosed at any point; and (iii) the method is purely a *property predictor*—it does not propose, generate, or enumerate novel molecules.

Toxicity false-negative risk. Misclassifying a toxic compound as safe is a concrete downstream harm. Users must treat model outputs as decision-support tools rather than regulatory verdicts, and must require experimental validation before advancing any compound, especially when predictions fall outside the training distribution. This caveat is particularly critical in OOD regimes, which are the primary focus of this work.

Data ethics and licensing. BOOM (Antoniuk et al., 2026) is a publicly released benchmark, and MoleculeNet (Wu et al., 2018) is an open MIT-licensed benchmark suite. The ChEMBL bioactivity data reach us through the Activity Cliffs benchmark (Van Tilborg et al., 2022) and are distributed under the EMBL-EBI / CC-BY licence. Pretrained GIN checkpoints follow Hu et al. (2019) (MIT licence) and UniMol checkpoints follow Zhou et al. (2023) (MIT licence). No proprietary or patient-level data are used.

Environmental cost. Experiments were run on RTX 3090 and A100 GPUs (see the implementation appendix); the retrieval-quality and displacement-probe analyses (Sections 4.3 and 4.4) used an H100. Total compute is approximately on the order of 10^3 GPU-hours.

Fairness and access. Drug-discovery benefits accrue unevenly across diseases and populations. We specifically encourage application of transductive few-shot methods—where our regime is most impactful—to neglected tropical diseases and rare-disease targets, where large labelled datasets are rarely available and computational triage is most needed.

Open-access commitment. Code and pretrained weights will be released under the MIT licence. An anonymised supplementary archive accompanies this submission to support reproducibility. We urge practitioners to pair these predictions with appropriate experimental validation and domain expertise before making any real-world deployment decision.

REPRODUCIBILITY STATEMENT

Every number from our own runs follows the protocol summarized in Section 4.1 and detailed in the appendices. Dataset sources, split construction, and the scaffold-overlap checks that define the X- and Y-split regimes are given in Section C. Encoder fine-tuning recipes, MALT hyperparameters, retrieval settings, and the memory-bank refresh schedule are in Section D and Table 7; the training and inference procedures, including the stop-gradient bank and the top-k retrieval step, are given as pseudocode in Section A, and the cost of those steps is quantified in Section F. Reported results are means over 5 seeds (3 for BOOM), and every MALT variant is initialized from the fine-tuned checkpoint of its matched inductive baseline, so that the two differ only in the prediction head and the retrieval step. The anonymized code release contains the code and configurations to fine-tune each encoder and to train and evaluate MALT-Fusion and MALT-Delta with it, together with the data-processing pipeline.

AI USE STATEMENT

We used generative AI tools in the course of this work and disclose their use below.

Research ideation, methodology, and literature review. The research question, the MALT framework, the design of the latent-fusion and latent-delta prediction heads, the choice of OOD regimes and benchmarks, and the experimental protocol were conceived and developed by the authors. We did not use generative AI tools to generate research ideas or hypotheses, to design the method, or to conduct the literature survey; all cited work was located and read by the authors.

Code and experiments. We used an AI coding assistant while implementing and running the experiments, including writing and refactoring training, evaluation, and analysis scripts, and orchestrating jobs on our compute cluster. All AI-assisted code was reviewed by the authors, and every number from our own experiments comes from runs on our infrastructure that the authors inspected; the BOOM baseline rows are quoted from Antoniuk et al. (2026).

Writing and manuscript preparation. We used AI assistants for language editing—grammar, concision, and consistency checking across sections—and for drafting and revising some passages, including result descriptions, table captions, and appendix text. The authors reviewed every such passage and verified all scientific claims, numerical results, and citations.

Data, results, and figures. We did not use generative AI to generate or alter experimental data. An AI coding assistant aggregated run outputs into tables, cross-checked reported numbers against the saved runs, and updated table entries from those outputs; the authors checked the resulting tables. All datasets are public benchmarks used as released, as described in Section C.

We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Evan Antoniuk, Shehtab Zaman, Tal Ben-Nun, Peggy Li, James Diffenderfer, Busra Sahin, Obadiah Smolenski, Everett Grethel, Tim Hsu, Anna Hiszpanski, et al. Boom: benchmarking out-of-distribution molecular property predictions of machine learning models. *Advances in Neural Information Processing Systems*, 38, 2026.

Sarp Aykent and Tian Xia. Gotennet: Rethinking efficient 3d equivariant graph neural networks. In *International Conference on Learning Representations*, volume 2025, pp. 40622–40646, 2025.

Wenxuan Bao, Zhichen Zeng, Zhining Liu, Hanghang Tong, and Jingrui He. Matcha: Mitigating graph structure shifts with test-time adaptation. In *International Conference on Learning Representations*, volume 2025, pp. 56906–56944, 2025.

Guanzi Chen, Jiying Zhang, Xi Xiao, and Yang Li. Graphtta: Test time adaptation on graph neural networks. *arXiv preprint arXiv:2208.09126*, 2022.

Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *IEEE Transactions on Big Data*, 2025.

Yury Gorishniy, Ivan Rubachev, Nikolay Kartashev, Daniil Shlenskii, Akim Kotelnikov, and Artem Babenko. Tabr: Tabular deep learning meets nearest neighbors. In *International Conference on Learning Representations*, volume 2024, pp. 18209–18249, 2024.

Esther Heid, Kevin P Greenman, Yunsie Chung, Shih-Cheng Li, David E Graff, Florence H Vermeire, Haoyang Wu, William H Green, and Charles J McGill. Chemprop: a machine learning package for chemical property prediction. *Journal of chemical information and modeling*, 64(1):9–17, 2024.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

Yuanfeng Ji, Lu Zhang, Jiaxiang Wu, Bingzhe Wu, Lanqing Li, Long-Kai Huang, Tingyang Xu, Yu Rong, Jie Ren, Ding Xue, et al. Drugood: Out-of-distribution dataset curator and benchmark for ai-aided drug discovery–a focus on affinity prediction problems with noise annotations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8023–8031, 2023.

Shengli Jiang, Shiyi Qin, Reid C Van Lehn, Prasanna Balaprakash, and Victor M Zavala. Uncertainty quantification for molecular property predictions with graph neural architecture search. *Digital Discovery*, 3(8):1534–1553, 2024.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. *arXiv preprint arXiv:1901.05761*, 2019.

Gang Liu, Srijit Seal, John Arevalo, Zhenwen Liang, Anne Carpenter, Meng Jiang, and Shantanu Singh. Learning molecular representation in a cell. In *International Conference on Learning Representations*, volume 2025, pp. 8540–8562, 2025.

Jiashuo Liu, Zheyan Shen, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.

Gerald M Maggiora. On outliers and activity cliffs why qsar often disappoints, 2006.

Aviv Netanyahu, Abhishek Gupta, Max Simchowitz, Kaiqing Zhang, and Pulkit Agrawal. Learning to extrapolate: A transductive approach. *arXiv preprint arXiv:2304.14329*, 2023.

M Oldenhof, G Ács, B Pejo, A Schuffenhauer, N Holway, N Sturm, A Dieckmann, O Fortmeier, E Boniface, C Mayer, et al. Industry-scale orchestrated federated learning for drug discovery. arxiv. *arXiv preprint arXiv:2210.08871*, 2022.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

Johannes Schimunek, Philipp Seidl, Lukas Friedrich, Daniel Kuhn, Friedrich Rippmann, Sepp Hochreiter, and Günter Klambauer. Context-enriched molecule representations improve few-shot drug discovery. *arXiv preprint arXiv:2305.09481*, 2023.

Nofit Segal, Aviv Netanyahu, Kevin P Greenman, Pulkit Agrawal, and Rafael Gómez-Bombarelli. Known unknowns: Out-of-distribution property prediction in materials and molecules. *npj Computational Materials*, 11(1):345, 2025.

Eduardo Soares, Emilio Vital Brazil, Victor Shirasuna, Dmitry Zubarev, Renato Cerqueira, and Kristin Schmidt. An open-source family of large encoder-decoder foundation models for chemistry. *Communications Chemistry*, 8(1):193, 2025.

Ava P Soleimany, Alexander Amini, Samuel Goldman, Daniela Rus, Sangeeta N Bhatia, and Connor W Coley. Evidential deep learning for guided molecular property prediction and discovery. *ACS central science*, 7(8):1356, 2021.

Prudencio Tossou, Cas Wognum, Michael Craig, Hadrien Mary, and Emmanuel Noutahi. Real-world molecular out-of-distribution: Specification and investigation. *Journal of Chemical Information and Modeling*, 64(3):697, 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, 2022.

Derek van Tilborg, Luke Rossen, and Francesca Grisoni. Molecular deep learning at the edge of chemical space. *Nature Machine Intelligence*, pp. 1–13, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

Runzhong Wang, Rui-Xi Wang, Mrunali Manjrekar, and Connor W Coley. Neural graph matching improves retrieval augmented generation in molecular machine learning. *arXiv preprint arXiv:2502.17874*, 2025.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 2526–2547, 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(2):513–530, 2018.

Ziting Xian, Jiawei Gu, Lingbo Li, and Shangsong Liang. Molrag: unlocking the power of large language models for molecular property prediction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15513–15531, 2025.

Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8): 3370, 2019.

Yiming Yang, Zhiyuan Zhou, Yueming Yin, Hoi-Yeung Li, and Adams Kong. Test-time adaptation without source data for out-of-domain bioactivity prediction. In *International Conference on Learning Representations*, volume 2026, pp. 4514–4532, 2026.

Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-mol: A universal 3d molecular representation learning framework. In *The eleventh international conference on learning representations*, 2023.

Xiang Zhuang, Qiang Zhang, Keyan Ding, Yatao Bian, Xiao Wang, Jingsong Lv, Hongyang Chen, and Huajun Chen. Learning invariant molecular representation in latent discrete space. *Advances in Neural Information Processing Systems*, 36:78435–78452, 2023.

SUPPLEMENTARY MATERIAL

A ALGORITHM OVERVIEW

Algorithm 1 MALT joint training (MALT-Fusion or MALT-Delta head)

Require: Training data $\mathcal{D}_{\text{train}} = \{(m_i, y_i)\}$; encoder $\mathcal{E}_\theta$; transduction module $\mathcal{T}$; head $\mathcal{P} \in \{\mathcal{P}_{\text{Fusion}}, \mathcal{P}_{\text{Delta}}\}$; task loss $\mathcal{L}_{\text{task}}$ (L1 for regression, BCE-with-logits for classification); refresh interval $N_{\text{update}} = 1$; auxiliary-loss weight $\lambda_\Delta \geq 0$ (Delta only)

Initialize: parameters $\theta_\mathcal{E}, \theta_\mathcal{T}, \theta_\mathcal{P}$; optimizer; scheduler
Build memory bank $\mathcal{B} \leftarrow \{(m_i,\ \bar{z}_i,\ y_i)\}$ where $\bar{z}_i = \text{sg}(\mathcal{E}_\theta(m_i))$ {no grad}
for epoch $= 1$ to MaxEpochs **do**
  if epoch > 1 and epoch $\bmod N_{\text{update}} = 0$ **then**
    Refresh $\bar{z}_i \leftarrow \text{sg}(\mathcal{E}_\theta(m_i))$ for all i {update bank with current $\mathcal{E}_\theta$}
  end if
  for each minibatch $(M_b,\ y_b) \subset \mathcal{D}_{\text{train}}$ **do**
    $z_q \leftarrow \mathcal{E}_\theta(M_b)$ {encode queries with grad}
    Mask self-entries; retrieve anchors $\{(\bar{z}_{a_j},\ y_{a_j})\}_{j=1}^{k} \leftarrow \mathcal{T}(z_q,\ \mathcal{B})$ {top-k Euclidean NN}
    if $\mathcal{P} = \mathcal{P}_{\text{Fusion}}$ **then**
      Compute cross-attention weights α_j^h (eq. 9) and attended rep z_{attn} (eq. 10)
      $\hat{y}_q \leftarrow W_{\text{out}}\,\sigma(W_{\text{fuse}}[z_q;\ z_{\text{attn}}])$ (eq. 11)
      $\mathcal{L} \leftarrow \mathcal{L}_{\text{task}}(\hat{y}_q,\ y_b)$
    else
      $\Delta z_j \leftarrow z_q - \bar{z}_{a_j}$; optionally append RDKit edit features e_j {$\mathcal{P}_{\text{Delta}}$; eq. 12}
      Compute bilinear correction $\hat{\mathbf{\Delta}}_j$ (eq. 13) and weights λ_j (eq. 14 or 15)
      $\hat{y}_q \leftarrow \sum_{j=1}^{k} \lambda_j\,(y_{a_j} + \hat{\mathbf{\Delta}}_j)$ (eq. 16)
      $\mathcal{L} \leftarrow \mathcal{L}_{\text{task}}(\hat{y}_q,\ y_b) + \lambda_\Delta \sum_{j=1}^{k} \text{sg}(\lambda_j)\,\left|\hat{\mathbf{\Delta}}_j - (y_b - y_{a_j})\right|$ {stop-grad weights; active when $\lambda_\Delta > 0$}
    end if
    Backpropagate $\mathcal{L}$; update $\theta_\mathcal{E}, \theta_\mathcal{T}, \theta_\mathcal{P}$ {end-to-end; stored $\bar{z}_i, y_i$ unchanged}
  end for
end for
return $\theta_\mathcal{E}, \theta_\mathcal{T}, \theta_\mathcal{P}$; rebuild $\mathcal{B}$ once with final $\mathcal{E}_\theta$ {inference bank}

Algorithm 2 MALT inference (MALT-Fusion or MALT-Delta head)

Require: Query molecule m_q; inference memory bank $\mathcal{B} = \{(m_i, \bar{z}_i, y_i)\}$; head $\mathcal{P} \in \{\mathcal{P}_{\text{Fusion}}, \mathcal{P}_{\text{Delta}}\}$; trained parameters $\theta_\mathcal{E}, \theta_\mathcal{T}, \theta_\mathcal{P}$

$z_q \leftarrow \mathcal{E}_\theta(m_q)$ {encode query}
$\{(\bar{z}_{a_j},\ y_{a_j})\}_{j=1}^{k} \leftarrow \mathcal{T}(z_q,\ \mathcal{B})$ {top-k via FAISS or PyTorch brute-force (App. F)}
if $\mathcal{P} = \mathcal{P}_{\text{Fusion}}$ **then**
  Compute α_j^h, z_{attn} (eqs. 9–10) {$\mathcal{P}_{\text{Fusion}}$; anchor labels y_{a_j} unused}
  $\hat{y}_q \leftarrow W_{\text{out}}\,\sigma(W_{\text{fuse}}[z_q;\ z_{\text{attn}}])$ (eq. 11)
else
  $\Delta z_j \leftarrow z_q - \bar{z}_{a_j}$; optionally compute RDKit edit features e_j {$\mathcal{P}_{\text{Delta}}$; uses anchor labels y_{a_j}}
  Compute $\hat{\mathbf{\Delta}}_j$ (eq. 13) and weights λ_j (eq. 14 or 15)
  $\hat{y}_q \leftarrow \sum_{j=1}^{k} \lambda_j\,(y_{a_j} + \hat{\mathbf{\Delta}}_j)$ (eq. 16)
end if
return $\hat{y}_q$

B Model Architecture

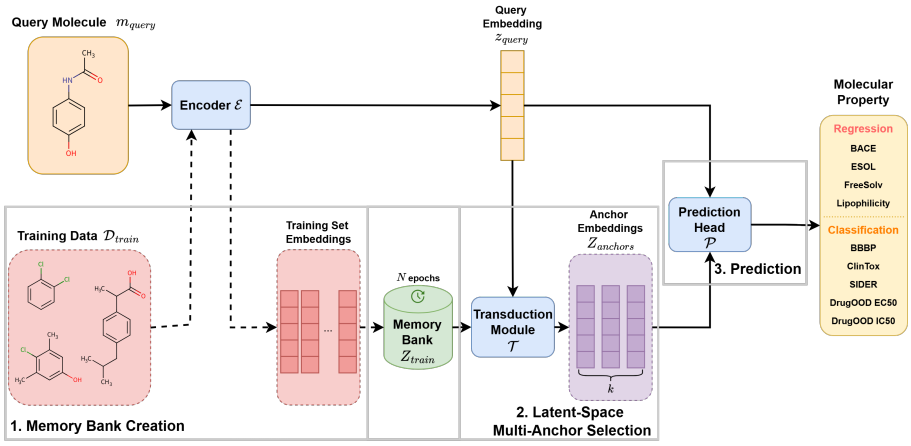
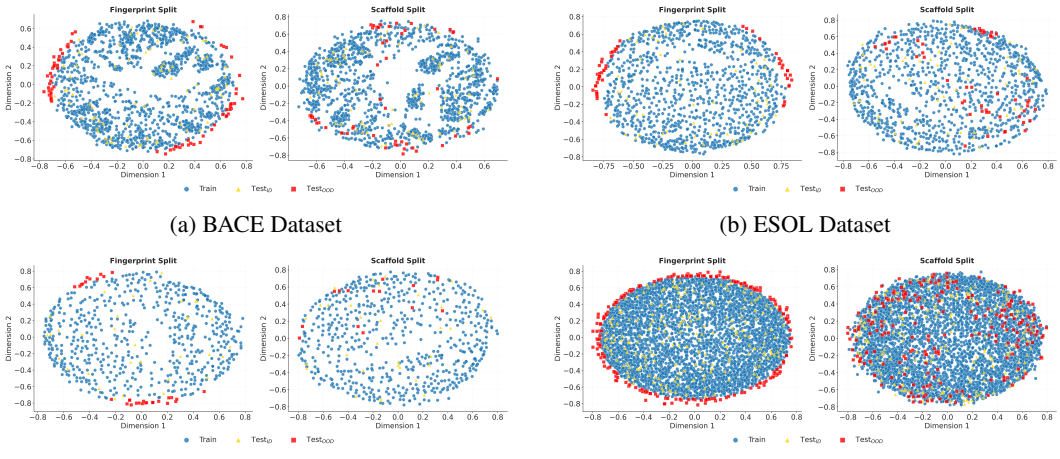


Figure 4: **MALT architecture.** A query molecule and the training molecules are encoded into a shared latent space. The memory bank stores stop-gradient training embeddings, top-k retrieval returns anchor embeddings, and the MALT-Fusion head predicts from query–anchor latent fusion. Anchor labels are stored for supervision and for MALT-Delta, but are not passed to the MALT-Fusion prediction head.

C Datasets and Split Methodology

We use scaffold-based splitting to construct structural OOD evaluation sets. Although fingerprint-based splits can appear to produce clearer separation in a 2D MDS projection of fingerprint Tanimoto similarity (Figure 5), this projection does not directly measure scaffold novelty.

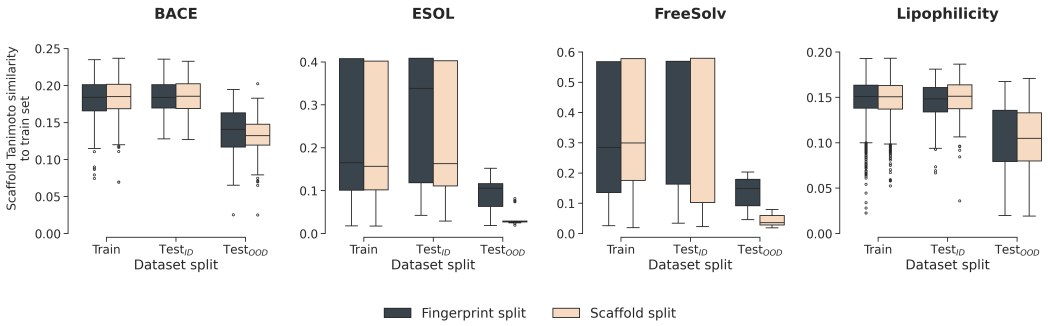
We therefore compare scaffold and fingerprint splits using three Bemis–Murcko scaffold-level similarity metrics: scaffold Tanimoto similarity, Maximum Common Substructure (MCS) similarity, and CATS pharmacophore similarity (Figure 6). Scaffold splits produce Test_{OOD} sets with lower scaffold-level similarity to the training set, especially under Tanimoto and MCS similarity. This supports our use of scaffold-based splits for structural OOD evaluation.



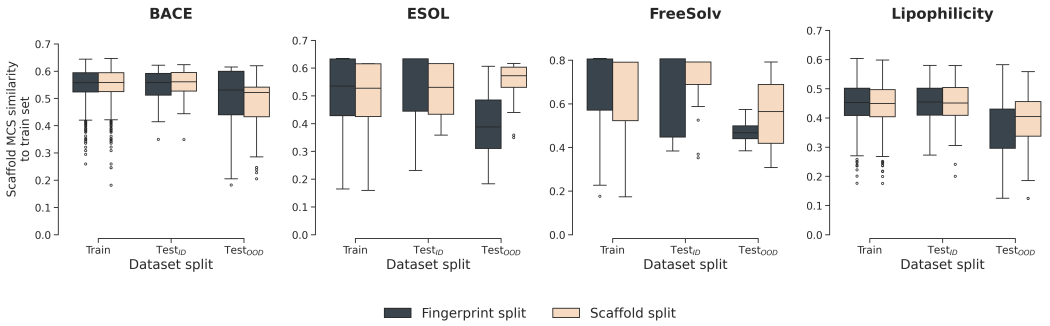
(a) BACE Dataset (b) ESOL Dataset

(c) FreeSolv Dataset (d) Lipophilicity Dataset

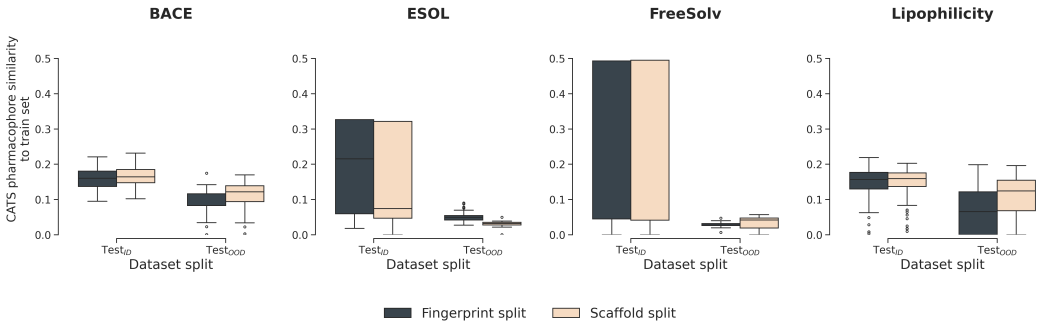
Figure 5: Chemical space distribution comparison for fingerprint and scaffold splits across four molecular property datasets. Each panel displays a 2D MDS projection based on molecular fingerprint Tanimoto similarity. Colors distinguish Train, Test_{ID}, and Test_{OOD} samples.



(a) Bemis–Murcko scaffold Tanimoto similarity.



(b) Bemis–Murcko scaffold Maximum Common Substructure (MCS) similarity.



(c) Bemis–Murcko scaffold CATS pharmacophore similarity.

Figure 6: Comparison of scaffold-level similarity between the test sets and the training set for fingerprint versus scaffold splits, evaluated using three Bemis–Murcko scaffold similarity criteria: (a) Tanimoto similarity, (b) Maximum Common Substructure similarity, and (c) CATS pharmacophore similarity. Scaffold splits generally yield lower similarity for Test_{OOD}, indicating stronger structural and functional distributional shifts.

D Implementation Details and Hyperparameters

We trained the original models on 4 × AMD EPYC 7742 CPUs, 8 × RTX 3090 GPUs, and 512 GB RAM; re-run experiments used A100 GPUs and the retrieval-quality and displacement-probe analyses (Sections 4.3 and 4.4) an H100. Hyperparameter configurations and training loss curves are reported in Table 7 and Figure 7; maximum epoch budgets are treated as upper bounds, with the training-loss patience rules below used to stop runs after convergence.

Inductive training procedure. All inductive baselines follow the reference fine-tuning recipe shipped with each encoder, so that comparisons isolate the inference-time operation rather than the training pipeline. Per-encoder details are as follows.

- **Chemprop** (Heid et al., 2024). We use the official `chemprop` package via its public Python API; Default training uses 30 epochs, the built-in optimizer and schedule, and MAE or cross-entropy losses selected from the task type.
- **Pretrained GIN** (Hu et al., 2019). The encoder is initialized from the public `supervised_contextpred.pth` checkpoint (5-layer GIN, emb_dim=300, mean pooling, JK=last, dropout 0.5). Fine-tuning uses Adam, learning rate 10^{-4}, batch size 64, 200 epochs, 50-epoch early-stopping patience, and no weight decay. The loss is MAE for regression and binary cross-entropy for classification.
- **UniMol** (Zhou et al., 2023). UniMol is trained via the official `unimol_tools` package and its `MolTrain` wrapper; Hyperparameters follow the package defaults: learning rate 5×10^{-4}, batch size 64, 100 epochs, 20-epoch patience, 5-fold cross-validation, and AdamW with linear warm-up.
- **RDKit descriptors.** We use 2048-bit Morgan fingerprints with radius 2 plus the default RDKit physicochemical descriptor set, followed by a 3-layer MLP with hidden size 512, ReLU activations, and dropout 0.2. Training uses Adam, learning rate 10^{-3}, batch size 128, 100 epochs, and 20-epoch patience.

For every encoder we use the seeds specified in Section 4.1: 5 seeds for MoleculeNet and Activity Cliffs, and 3 seeds for BOOM.

Transductive adaptation procedure. MALT-Fusion and MALT-Delta are trained as transductive adaptations on top of the fine-tuned inductive checkpoints. They reuse the matched encoder initialization and task loss, replace the inductive prediction head with the transductive head, and optimize the encoder and head jointly while retrieving from a stop-gradient training bank. The official inductive pipelines use validation splits for model selection and early stopping: for Y-splits, validation molecules are sampled randomly from the non-OOD training-support region; for X-splits, validation is chosen after scaffold splitting from the training scaffolds in Section C. The transductive adaptation does not introduce a separate transductive validation set. Following Segal et al. (2025), test molecules are visible only as unlabeled retrieval queries, test labels are never used, and no parameters are updated at inference. Consequently, transductive training uses a fixed maximum epoch budget with a training-loss patience rule only; this stopping rule is not an OOD model-selection signal.

For the reported Chemprop and GIN MALT-Fusion and MALT-Delta runs, the maximum budget is 500 epochs with 50-epoch training-loss patience, batch size 64, learning rate 10^{-4}, exponential decay γ=0.999, and memory-bank refresh every epoch. For UniMol, MALT-Fusion uses 100 epochs with 30-epoch training-loss patience, batch size 32, learning rate 10^{-4}, and γ=0.97; the protocol-matched MALT-Delta-UniMol setting uses 500 epochs with 50-epoch patience, batch size 16, learning rate 10^{-4}, and γ=0.999 because of UniMol memory constraints. All settings use k=10 retrieved anchors for main results unless otherwise stated.

Table 7: Hyperparameter ranges considered. Final settings are given in the text; some lie outside these ranges (inductive GIN: 200 epochs; UniMol: inductive learning rate 5×10^{-4}, MALT batch size 32 / MALT-Delta 16).

Parameters	**Settings**	**Values**
Batch Size		64, 128, 256, 512
Epochs	Inductive Models	10, 30, 50, 100
	BLT, MALT (regression)	100, 200, 500, 1000
	BLT (classification)	100, 500, 1000, 2000
	MALT (classification)	10, 20, 50, 100
Number of anchors k		1, 3, 5, 10
Learning Rate		10^{-3}, 10^{-4}, 10^{-5}

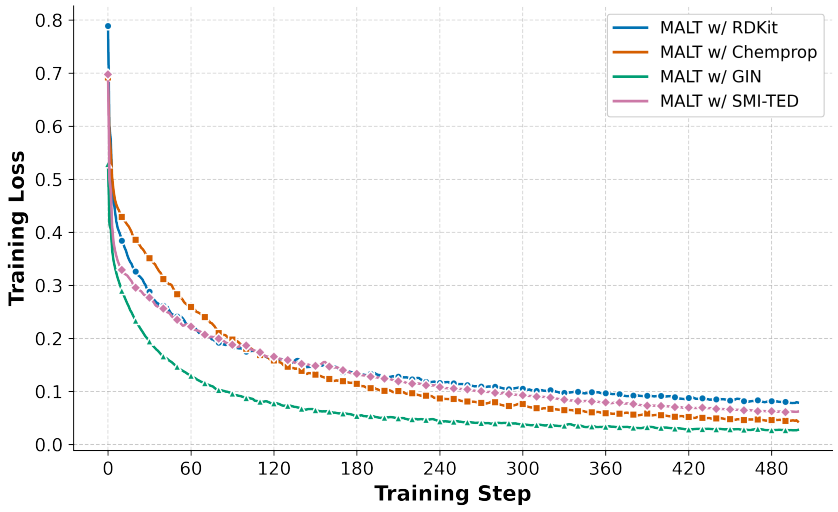


Figure 7: Comparison of training loss curves for MALT variants using RDKit, Chemprop, and GIN representations across 500 training steps.

E Additional Experimental Results

This appendix follows the same hierarchy as Section 4: anchor quality, retrieval controls, mechanism controls, benchmark details, visual diagnostics, robustness, and compute overhead.

E.1 Anchor Case Studies

To evaluate the effectiveness of multi-anchor reasoning within learned latent spaces, we keep this section to three representative ESOL case studies and compare the top-5 anchors selected by MALT and BLT after training. As shown in Figure 8, we analyze three representative ESOL molecules from Test_{OOD}: (i) a random query, (ii) an extreme covariate-shift query with the lowest average Maximum Common Substructure (MCS) similarity to training scaffolds, and (iii) an extreme label-shift query with the highest target value.

For each case, we assess how well the retrieved anchors align with the query in both target value and MCS similarity. MALT consistently selects anchors that are more functionally and structurally aligned with the query than BLT's. BLT's anchors are often farther away in both chemistry and property space. The contrast is especially clear in the extreme covariate-shift example, where the learned latent space recovers subtle structural analogies that descriptor-space matching misses. In the extreme label-shift case, no single training molecule is sufficient; MALT benefits from combining multiple complementary anchors, whereas BLT's single-anchor mechanism is more brittle.

E.2 Aggregate Anchor Chemistry

To quantify the anchor selection mechanism, we compare the training data, the OOD queries, and the anchors selected by MALT. We examine physicochemical properties, structural motifs, and similarity metrics to expose the chemical regularities that govern retrieval.

E.2.1 Physicochemical Property Analysis

We compare three core properties: molecular weight (MW), logarithm of the partition coefficient (LogP), and topological polar surface area (TPSA). The results in Table 8 cover both scaffold-based (X-split) and property-based (Y-split) OOD settings. For most datasets, selected anchors lie between the training distribution and the OOD queries (15 of 24 property means), but for FreeSolv-X and Lipo-X they stay near the training distribution.

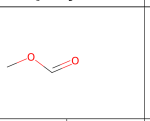
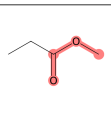
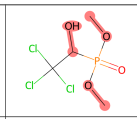
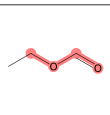
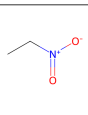
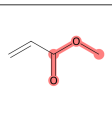
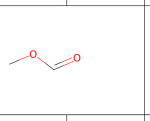
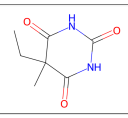
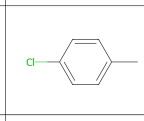
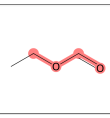
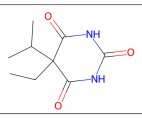
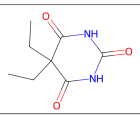
Table 8: Physicochemical property comparison. For each dataset, results are presented for scaffold-based (X-split, left) and property-based (Y-split, right) OOD settings.

		X-Split			Y-Split		
Dataset	**Property**	**Train** *Mean (Median)*	**OOD (Query)** *Mean (Median)*	**Anchors** *Mean (Median)*	**Train** *Mean (Median)*	**OOD (Query)** *Mean (Median)*	**Anchors** *Mean (Median)*
BACE	Mol. Weight	481.7 (465.6)	447.1 (452.0)	476.3 (422.5)	472.3 (457.5)	617.4 (608.7)	579.1 (579.8)
	LogP	3.11 (3.12)	3.62 (4.60)	3.31 (3.52)	3.12 (3.13)	3.10 (3.30)	3.36 (3.37)
	TPSA	95.3 (91.2)	97.5 (78.2)	101.1 (85.6)	93.5 (89.6)	128.3 (111.9)	113.1 (108.3)
ESOL	Mol. Weight	196.6 (179.2)	337.2 (307.3)	275.9 (268.4)	209.7 (192.0)	108.5 (88.1)	145.1 (108.1)
	LogP	2.38 (2.30)	3.58 (4.17)	3.18 (3.40)	2.58 (2.44)	0.07 (0.39)	0.61 (1.01)
	TPSA	33.7 (26.0)	56.4 (56.7)	46.5 (41.6)	34.8 (26.3)	42.1 (26.6)	48.4 (28.7)
FreeSolv	Mol. Weight	134.8 (120.2)	258.6 (241.3)	123.9 (118.2)	141.0 (122.1)	102.9 (100.2)	111.3 (99.0)
	LogP	1.88 (1.74)	3.83 (3.75)	1.68 (1.56)	1.91 (1.75)	2.71 (2.76)	2.29 (2.04)
	TPSA	20.3 (17.1)	18.7 (9.2)	21.5 (20.2)	21.0 (18.5)	0.0 (0.0)	0.2 (0.0)
Lipo	Mol. Weight	387.2 (390.9)	320.6 (305.8)	404.8 (401.9)	381.4 (386.5)	417.1 (423.5)	419.6 (427.4)
	LogP	3.31 (3.31)	2.71 (2.37)	3.29 (3.22)	3.22 (3.22)	4.39 (4.25)	3.90 (3.68)
	TPSA	79.8 (80.7)	65.6 (63.2)	81.6 (83.1)	79.4 (80.0)	69.6 (69.2)	75.8 (76.1)

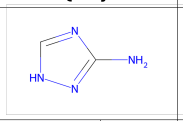
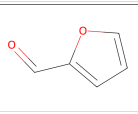
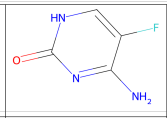
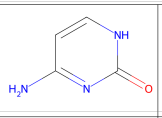
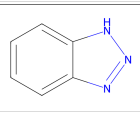
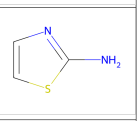
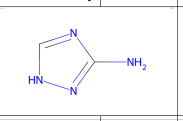
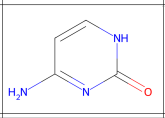
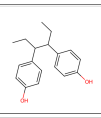
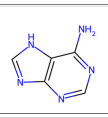
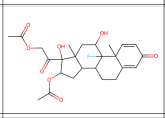
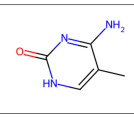
E.2.2 Structural and Fragment Analysis

To probe the structural basis of anchor selection, we analyze Murcko scaffolds, BRICS motifs, and RECAP fragments. Tables 9 and 10 show that selected anchors share more chemically relevant motifs with the OOD queries than a random sample from the training set would, supporting the view that anchor retrieval is grounded in structural relevance rather than superficial feature overlap.

(a) Random

	Query		Top 5 Anchors				
Ours							
	Property	1.6678	1.3194	1.2807	1.4597	1.2807	1.2807
	MCS similarity	1.0000	0.6364	0.0559	0.7778	0.0000	0.6364
BLT							
	Property	1.6678	0.7930	-0.1031	1.4597	0.3178	0.2259
	MCS similarity	1.0000	0.0000	0.0000	0.7778	0.0000	0.0000

(b) Extreme Covariate Shift

	Query		Top 5 Anchors				
Ours							
	Property	1.6397	1.3388	0.9169	0.8283	1.0098	1.2130
	MCS similarity	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
BLT							
	Property	1.6397	0.8283	-0.7563	0.3614	-0.6112	0.6817
	MCS similarity	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000

(c) Extreme Label Shift

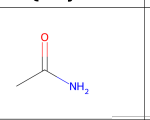
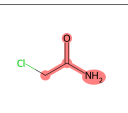
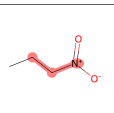
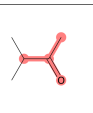
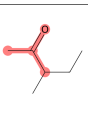
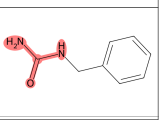
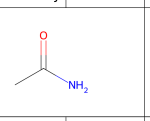
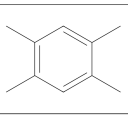
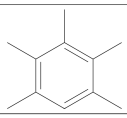
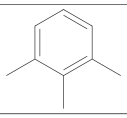
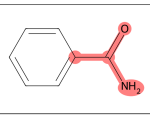
	Query		Top 5 Anchors				
Ours							
	Property	2.1517	1.3775	1.0001	1.3291	1.0630	0.9275
	MCS similarity	1.0000	0.7778	0.3247	0.3247	0.2747	0.1623
BLT							
	Property	2.1517	-0.8337	-0.5483	-0.1612	0.9227	0.7872
	MCS similarity	1.0000	0.0000	0.0000	0.0000	0.3889	0.0000

Figure 8: Top-5 anchors retrieved by MALT vs. BLT on ESOL. Red highlights mark the Maximum Common Substructure (MCS) shared by the query and each anchor. We show a random OOD query, an extreme covariate-shift query, and an extreme label-shift query.

Table 9: Top scaffolds and structural diversity metrics for the X-split.

Dataset	Data Type	Top 1 Scaffold (Count)	Top 2 Scaffold (Count)	Entropy	Diversity
BACE	Train	O=C1NC=NC1(c1ccccc1)c1ccccc1 (52)	O=S1(=O)CC(Cc2ccccc2)CC([NH2+]Cc2ccccc2)C1 (50)	8.327	0.445
	OOD	c1ccc(-c2ccc(-c3ccccc3)n2Cc2cccn2)cc1 (14)	c1ccccc1 (7)	4.173	0.527
	Anchors	C1=NC2(CO1)c1ccccc1Oc1ccc(-c3cccc3)cc12 (105)	O=S1(=O)CC(Cc2ccccc2)CC([NH2+]Cc2ccccc2)C1 (28)	5.234	0.169
ESOL	Train	No Scaffold (303)	c1ccccc1 (234)	4.797	0.217
	OOD	O=C(OCc1cccc(Oc2ccccc2)c1)C1CC1 (7)	O=c1oc2ccccc2cc1Cc1ccccc1 (2)	5.280	0.833
	Anchors	c1ccccc1 (77)	O=C1CC(=O)NC(=O)N1 (51)	5.297	0.168
FreeSolv	Train	No Scaffold (305)	c1ccccc1 (143)	2.656	0.079
	OOD	c1ccc(Cn2ccnc2)cc1 (1)	c1ccc(Cc2ccccc2)cc1 (1)	3.907	1.000
	Anchors	No Scaffold (92)	c1ccccc1 (31)	1.901	0.080
Lipo	Train	c1ccc(-c2ccccc2)cc1 (31)	O=C(Cc1ccccc1)NC1CCN(CCC(c2ccccc2)c2ccccc2)CC1 (28)	10.539	0.587
	OOD	c1ccccc1 (76)	c1ccncc1 (6)	5.000	0.462
	Anchors	c1cnc(-c2ccc(C3CCCCC3)cc2)cn1 (53)	O=S(=O)(NCC(c1ccccc1)N1CCCCCC1)c1ccccc1 (47)	8.373	0.291

Table 10: Analysis of common BRICS motifs and RECAP fragments for the X-split.

		BRICS Analysis				RECAP Analysis			
Dataset	Data Type	Top 1 (Count)	Top 2 (Count)	Diversity	Entropy	Top 1 (Count)	Top 2 (Count)	Diversity	Entropy
BACE	Train	*N* (729)	*c1cccc(*)c1 (655)	-	-	*C(C)=O (205)	*S(C)(=O)=O (141)	0.323	8.641
	OOD	*C* (43)	*c1ccc(*)cc1 (29)	-	-	*c1ccc(*)n1* (25)	*Cc1cccc(N)n1 (20)	0.382	5.403
	Anchors	*N* (262)	*c1ccccc1 (247)	-	-	*c1ccccc1 (122)	*c1ccc(*)n1* (121)	0.114	6.431
ESOL	Train	*CC (131)	*O* (105)	0.447	8.155	*C(C)C (30)	*C(C)=O (29)	0.474	7.614
	OOD	*c1ccccc1 (16)	*O* (14)	0.503	5.761	*c1ccccc1 (10)	*CC1C(C(*)=O)C1(C)C (6)	0.600	5.218
	Anchors	*O* (97)	*CC (71)	0.179	6.582	*C(C)=O (39)	*C(C)C (28)	0.280	6.492
FreeSolv	Train	*CC (49)	*OC (45)	0.578	7.991	*C(C)=O (16)	*OC (15)	0.460	6.249
	OOD	*c1ccccc1 (4)	*C* (2)	0.848	4.681	*O (2)	*Cc1ccccc1 (1)	0.917	3.418
	Anchors	*O* (22)	*CC (15)	0.398	5.943	*C[C@H](C)O (6)	*CCCC (6)	0.508	4.571
Lipo	Train	*N* (2018)	*C(*)=O (1225)	0.116	7.642	*O (434)	*c1ccccc1 (335)	0.286	9.959
	OOD	*N* (62)	*C(*)=O (52)	0.294	6.715	*O (18)	*N1CCN(*)CC1 (16)	0.556	7.508
	Anchors	*N* (1093)	*c1ccccc1 (693)	0.071	7.019	*O (219)	*c1ccccc1 (206)	0.175	8.778

E.2.3 Quantitative Similarity Comparison

Finally, we quantify similarity between OOD queries and their selected anchors using Tanimoto similarity over ECFP4 fingerprints. We compare this against the similarity between the same queries and non-anchor training molecules. Table 11 shows that selected anchors are more similar to the queries than non-anchors in 15 of 16 comparisons, with the largest gaps at the scaffold level under label shift.

Table 11: Tanimoto similarity between OOD queries and training molecules' Morgan fingerprints. X and Y denote scaffold and label splits, respectively.

		BACE		**ESOL**		**FreeSolv**		**Lipo**	
Similarity Type	**Comparison**	X	Y	X	Y	X	Y	X	Y
Whole Molecule	Query vs. Anchor	**0.495**	**0.835**	**0.382**	**0.187**	0.048	**0.262**	**0.313**	**0.532**
	Query vs. Non-anchor	0.328	0.438	0.094	0.046	0.054	0.067	0.220	0.291
Scaffold	Query vs. Anchor	**0.351**	**0.808**	**0.314**	**0.567**	**0.075**	**0.762**	**0.180**	**0.516**
	Query vs. Non-anchor	0.193	0.295	0.098	0.020	0.060	0.008	0.116	0.216

These analyses reveal two global trends:

1. **Anchors are substantially more similar to queries than non-anchors.** This holds in 15 of 16 comparisons; the exception is FreeSolv-X at the whole-molecule level (0.048 vs. 0.054).
2. **Under label shift, the similarity gap is largest for scaffolds.** MALT identifies molecules with chemically relevant core structures, which supports more faithful extrapolation under OOD shift.

Table 12: Anchor Selection Strategy Comparison (MAE ↓). Each cell shows $\frac{\text{ID}}{\text{OOD}}$. Rank_{ID} and Rank_{OOD} average across both shifts; Overall averages all four settings. Best overall in **bold**.

		Covariate Shift				Label Shift						
Strategy	**Metric**	BACE	ESOL	FreeS.	Lipo	BACE	ESOL	FreeS.	Lipo	$\mathbf{R_{ID}}$	$\mathbf{R_{OOD}}$	$\mathbf{R_{All}}$
Adaptive	Cos.	$\frac{.305}{.685}$	$\frac{.226}{.567}$	$\frac{.179}{.375}$	$\frac{.357}{.533}$	$\frac{.399}{.739}$	$\frac{.207}{.501}$	$\frac{.154}{.346}$	$\frac{.314}{.503}$	9.0	**5.6**	7.3
	Cos.+w	$\frac{.311}{.751}$	$\frac{.223}{.554}$	$\frac{.182}{.372}$	$\frac{.350}{.514}$	$\frac{.398}{.749}$	$\frac{.206}{.496}$	$\frac{.152}{.339}$	$\frac{.312}{.528}$	8.6	7.1	7.9
	Euc.	$\frac{.306}{.712}$	$\frac{.201}{.572}$	$\frac{.181}{.382}$	$\frac{.345}{.538}$	$\frac{.411}{.747}$	$\frac{.207}{.510}$	$\frac{.142}{.370}$	$\frac{.314}{.484}$	7.9	9.3	8.6
	Euc.+w	$\frac{.396}{.690}$	$\frac{.215}{.594}$	$\frac{.204}{.382}$	$\frac{1.07}{1.74}$	$\frac{.452}{.493}$	$\frac{.225}{.456}$	$\frac{.146}{.354}$	$\frac{.935}{1.10}$	14.0	8.8	11.4
Temp.Exp.	Cos.	$\frac{.308}{.698}$	$\frac{.198}{.554}$	$\frac{.204}{.402}$	$\frac{.350}{.531}$	$\frac{.404}{.753}$	$\frac{.204}{.515}$	$\frac{.142}{.357}$	$\frac{.307}{.508}$	7.3	9.6	8.4
	Cos.+w	$\frac{.307}{.729}$	$\frac{.198}{.548}$	$\frac{.200}{.389}$	$\frac{.354}{.531}$	$\frac{.408}{.756}$	$\frac{.202}{.515}$	$\frac{.134}{.343}$	$\frac{.313}{.509}$	6.8	9.5	8.1
	Euc.	$\frac{.306}{.696}$	$\frac{.202}{.539}$	$\frac{.204}{.399}$	$\frac{.351}{.534}$	$\frac{.403}{.753}$	$\frac{.202}{.519}$	$\frac{.140}{.355}$	$\frac{.308}{.510}$	6.4	9.6	8.0
	Euc.+w	$\frac{.306}{.707}$	$\frac{.203}{.539}$	$\frac{.204}{.399}$	$\frac{.349}{.534}$	$\frac{.404}{.757}$	$\frac{.202}{.517}$	$\frac{.140}{.357}$	$\frac{.308}{.505}$	**6.1**	9.7	7.9
Temp.Lin.	Cos.	$\frac{.306}{.699}$	$\frac{.198}{.548}$	$\frac{.206}{.401}$	$\frac{.352}{.534}$	$\frac{.405}{.759}$	$\frac{.204}{.518}$	$\frac{.142}{.354}$	$\frac{.307}{.508}$	7.4	10.3	8.9
	Cos.+w	$\frac{.308}{.724}$	$\frac{.198}{.547}$	$\frac{.199}{.389}$	$\frac{.353}{.529}$	$\frac{.409}{.753}$	$\frac{.204}{.515}$	$\frac{.142}{.342}$	$\frac{.314}{.513}$	8.9	8.9	8.9
	Euc.	$\frac{.306}{.702}$	$\frac{.202}{.540}$	$\frac{.205}{.400}$	$\frac{.350}{.534}$	$\frac{.401}{.759}$	$\frac{.203}{.513}$	$\frac{.138}{.353}$	$\frac{.308}{.511}$	6.4	10.1	8.3
	Euc.+w	$\frac{.301}{.714}$	$\frac{.201}{.540}$	$\frac{.202}{.402}$	$\frac{.359}{.539}$	$\frac{.408}{.756}$	$\frac{.204}{.513}$	$\frac{.145}{.335}$	$\frac{.315}{.509}$	9.5	9.9	9.7
Top-k	Cos.	$\frac{.310}{.699}$	$\frac{.217}{.575}$	$\frac{.185}{.380}$	$\frac{.357}{.538}$	$\frac{.408}{.745}$	$\frac{.200}{.505}$	$\frac{.147}{.332}$	$\frac{.315}{.501}$	10.4	7.0	8.7
	Cos.+w	$\frac{.311}{.697}$	$\frac{.205}{.560}$	$\frac{.188}{.389}$	$\frac{.350}{.513}$	$\frac{.407}{.762}$	$\frac{.203}{.496}$	$\frac{.151}{.315}$	$\frac{.307}{.531}$	8.3	7.6	7.9
	Euc.	$\frac{.305}{.686}$	$\frac{.219}{.567}$	$\frac{.187}{.373}$	$\frac{.352}{.544}$	$\frac{.402}{.752}$	$\frac{.197}{.507}$	$\frac{.141}{.322}$	$\frac{.315}{.493}$	6.8	6.1	**6.5**
	Euc.+w	$\frac{.389}{.682}$	$\frac{.217}{.567}$	$\frac{.190}{.382}$	$\frac{1.07}{1.75}$	$\frac{.432}{.607}$	$\frac{.229}{.472}$	$\frac{.135}{.297}$	$\frac{.956}{1.35}$	12.3	6.9	9.6

E.3 Geometry and Retrieval Ablations

E.3.1 Anchor Selection Methods

We evaluate anchor selection strategies along three axes: selection mechanism (deterministic vs. stochastic), density adaptation, and distance metric. For each configuration, we also test whether incorporating retrieval distances as explicit input features, denoted "+w", improves prediction.

Selection strategies.

- **Top-k:** deterministically selects the k nearest neighbors in latent space.
- **Adaptive:** dynamically adjusts the effective anchor count $k_{\text{eff}} \in [k_{\min}, k_{\max}]$ based on local sample density, masking distant anchors in sparse regions.
- **Temperature sampling:** stochastically samples anchors from $p_i \propto \exp(-d_i/\tau)$, where d_i denotes the distance to anchor i. We evaluate both exponential and linear temperature schedules.

Distance metrics. We compare Euclidean distance and cosine similarity for retrieval and attention computation.

Results. Table 12 reports MAE across four MoleculeNet datasets under covariate and label shift. We rank methods by ID performance, OOD performance, and their overall average.

Top-k Euclidean gives the best overall rank and is the only setting that remains consistently strong across both ID and OOD regimes. Adaptive and temperature sampling variants can improve one regime while hurting the other. Adding explicit distance features ("+w") is unstable across datasets, especially on Lipophilicity.

Default configuration. We therefore use deterministic top-k retrieval with Euclidean distance.

E.3.2 Further Ablation Study Results

Table 13: Performance (MAE) Comparison of Encoder Training Strategies. Strategies involve inductive finetuning ("Finetune O/X") and whether the encoder is adapted during transduction ("Adapt O") or kept frozen ("Adapt X"). Cell colors in orange indicate settings where "Finetune O, Adapt O" shows lower MAE. Green indicates when other settings result in lower MAE. Bold: lowest MAE per model within each column.

		BACE		ESOL		FreeSolv		Lipophilicity	
Setting	**Models**	**Test$_{ID}$**	**Test$_{OOD}$**	**Test$_{ID}$**	**Test$_{OOD}$**	**Test$_{ID}$**	**Test$_{OOD}$**	**Test$_{ID}$**	**Test$_{OOD}$**
Covariate Shift (X-Splits)									
Finetune X, Adapt X	Chemprop	0.4508 ±0.0321	1.0973 ±0.1283	0.2995 ±0.0265	0.6525 ±0.0569	0.2145 ±0.0250	0.5827 ±0.1009	0.5273 ±0.0133	0.8090 ±0.0330
	Pretrained GNN	0.3212 ±0.0074	0.6763 ±0.0624	0.2612 ±0.0082	0.5326 ±0.0031	0.2120 ±0.0047	0.4389 ±0.0380	0.3648 ±0.0023	0.5606 ±0.0102
	SMI-TED light	0.3767 ±0.0171	0.8553 ±0.0431	0.3187 ±0.0182	0.5508 ±0.0156	0.2878 ±0.0204	0.5210 ±0.0461	0.4036 ±0.0076	0.5584 ±0.0160
Finetune O, Adapt X	Chemprop	0.3944 ±0.0183	0.9481 ±0.0341	**0.2061** ±0.0183	**0.4676** ±0.0423	**0.1487** ±0.0124	0.3423 ±0.0498	0.3758 ±0.0160	0.5545 ±0.0199
	Pretrained GNN	**0.2991** ±0.0097	0.6810 ±0.0712	**0.2101** ±0.0116	0.5310 ±0.0113	**0.1868** ±0.0103	0.3694 ±0.0362	0.3517 ±0.0100	0.5478 ±0.0064
	SMI-TED light	0.3342 ±0.0192	**0.6487** ±0.0390	0.2363 ±0.0115	**0.4141** ±0.0175	0.1989 ±0.0076	0.3272 ±0.0352	0.3976 ±0.0021	0.5584 ±0.0160
Finetune O, Adapt O	Chemprop	**0.2847** ±0.0165	**0.7783** ±0.0553	0.2180 ±0.0049	0.5072 ±0.0154	0.1522 ±0.0043	**0.2999** ±0.0164	**0.3474** ±0.0135	**0.4894** ±0.0263
	Pretrained GNN	0.3317 ±0.0083	**0.6333** ±0.0347	0.2103 ±0.0113	**0.5305** ±0.0120	0.1919 ±0.0126	**0.3388** ±0.0255	**0.3370** ±0.0007	**0.5369** ±0.0007
	SMI-TED light	**0.3037** ±0.0046	0.6716 ±0.0569	**0.2057** ±0.0052	0.4322 ±0.0118	**0.1497** ±0.0161	**0.2613** ±0.0329	**0.3608** ±0.0120	**0.5417** ±0.0164
Label Shift (Y-Splits)									
Finetune X, Adapt X	Chemprop	0.4572 ±0.0263	1.0237 ±0.1289	0.2594 ±0.0229	0.5097 ±0.0229	0.3075 ±0.0270	0.3461 ±0.0525	0.4991 ±0.0305	0.9026 ±0.0708
	Pretrained GNN	0.3999 ±0.0057	**0.7052** ±0.0254	0.2293 ±0.0084	0.5607 ±0.0134	0.2124 ±0.0269	0.3453 ±0.0156	0.3479 ±0.0113	0.5004 ±0.0068
	SMI-TED light	0.4104 ±0.0139	0.9685 ±0.0343	0.3686 ±0.0068	0.8628 ±0.0175	0.2868 ±0.0163	0.8585 ±0.0448	0.5246 ±0.0052	1.1654 ±0.0363
Finetune O, Adapt X	Chemprop	0.4132 ±0.0325	0.9366 ±0.0706	0.1941 ±0.0079	**0.4173** ±0.0284	0.2019 ±0.0173	**0.2009** ±0.0516	0.3489 ±0.0141	**0.5797** ±0.0571
	Pretrained GNN	0.4057 ±0.0111	0.7464 ±0.0158	0.1849 ±0.0090	0.5000 ±0.0283	**0.1386** ±0.0151	0.3516 ±0.0186	0.3497 ±0.0767	0.4903 ±0.0051
	SMI-TED light	0.3592 ±0.0092	0.9228 ±0.0226	0.2167 ±0.0140	0.5566 ±0.0281	0.2493 ±0.0245	0.4763 ±0.0395	0.3480 ±0.0113	0.7379 ±0.0206
Finetune O, Adapt O	Chemprop	**0.3461** ±0.0097	**0.7705** ±0.0287	**0.1734** ±0.0070	0.4994 ±0.0114	**0.1469** ±0.0179	0.2753 ±0.0530	**0.3185** ±0.0104	0.6444 ±0.0151
	Pretrained GNN	**0.3861** ±0.0097	0.7340 ±0.0121	**0.1845** ±0.0093	**0.4989** ±0.0267	0.1585 ±0.0165	**0.2637** ±0.0220	**0.3195** ±0.0108	**0.4690** ±0.0108
	SMI-TED light	**0.3546** ±0.0164	**0.8326** ±0.0141	**0.1852** ±0.0096	**0.5390** ±0.0199	**0.1716** ±0.0061	**0.2856** ±0.0427	**0.3300** ±0.0097	**0.6609** ±0.0331

Table 14: Noisy-anchor stress test for MALT-Fusion-GIN (MAE). At evaluation, n of the k=10 top-ranked anchors are replaced by the lowest-ranked ones. Rows come from a separate model run, so the n=0 row differs from Table 2. At n=5, label-shift OOD MAE changes by -6.6% to $+1.5\%$; under covariate shift, BACE OOD MAE rises from 0.7290 to 0.9222 at n=3 ($+26.5\%$).

	BACE		**ESOL**		**FreeSolv**		**Lipophilicity**	
Noisy Anchors	**Test$_{\text{ID}}$**	**Test$_{\text{OOD}}$**	**Test$_{\text{ID}}$**	**Test$_{\text{OOD}}$**	**Test$_{\text{ID}}$**	**Test$_{\text{OOD}}$**	**Test$_{\text{ID}}$**	**Test$_{\text{OOD}}$**
Covariate Shift (X-Splits)								
0 (Default)	0.3338	0.7290	0.2142	0.5572	0.2085	0.4477	0.3003	0.4736
1	0.3583	0.6343	0.2281	0.5683	0.2133	0.4395	0.2992	0.4713
3	0.3338	0.9222	0.2158	0.6063	0.2206	0.4603	0.3003	0.4746
5	0.3269	0.8346	0.2186	0.5853	0.2175	0.4486	0.3009	0.4748
Label Shift (Y-Splits)								
0 (Default)	0.4191	0.7115	0.2160	0.5395	0.2086	0.4235	0.3024	0.4743
1	0.4206	0.6879	0.2264	0.5146	0.2139	0.4074	0.3019	0.4746
3	0.4344	0.7271	0.2287	0.5279	0.2207	0.4393	0.3020	0.4736
5	0.4061	0.7068	0.2264	0.5041	0.2181	0.4298	0.3027	0.4747

Table 15: Effect of anchor count k on Test$_{\text{OOD}}$ MAE under label shift. Lower is better; best in **bold**.

k	BACE	ESOL	FreeSolv	Lipophilicity
1	0.751	0.542	0.400	0.586
3	0.759	0.504	0.372	0.471
5	0.761	0.504	0.357	0.484
10	**0.734**	**0.499**	**0.264**	**0.469**

E.4 Fixed-Neighbor Label Aggregation Controls

Table 16 compares MALT latent fusion with label-space aggregation under fixed retrieved neighbors. Latent fusion is best on six of the eight task/split cells, and the margin is largest when the embedding geometry is frozen rather than jointly trained. Two cells are exceptions: on BACE under label shift a 1-NN readout is better (0.73 within the MALT-joint group and 0.68 for ECFP, against 0.74 for latent fusion), and on FreeSolv Test_{OOD} attentive label readout reaches 0.33 against 0.34. The overall pattern still supports the claim that MALT's gains are not explained by retrieval alone or by direct neighbor-label readout.

Table 16: Full label-aggregation vs. embedding-fusion comparison (MAE↓, label shift). **Bold**: best rule within the MALT-joint group.

		Test_{OOD}				Test_{ID}			
Embedding	**Method**	BACE	ESOL	FSolv	Lipo	BACE	ESOL	FSolv	Lipo
MALT (joint)	**MALT (Ours)**	0.74	**0.52**	0.34	**0.49**	**0.36**	**0.18**	**0.19**	**0.33**
	Nadaraya–Watson	0.91	0.72	0.50	0.83	0.39	0.23	0.24	0.38
	Scaled Dot-Prod.	0.93	0.72	0.50	0.83	0.39	0.23	0.25	0.38
	IDW	0.90	0.70	0.48	0.81	0.39	0.22	0.22	0.36
	Attentive LR	0.87	0.59	**0.33**	0.62	0.38	0.24	0.27	0.34
	k-NN Avg	0.94	0.73	0.51	0.83	0.39	0.24	0.25	0.38
	1-NN	**0.73**	0.59	0.39	0.71	0.45	0.28	0.29	0.43
Pretrained GIN (frozen)	Nadaraya–Watson	1.22	1.22	1.39	1.34	0.57	0.47	0.60	0.70
	Scaled Dot-Prod.	1.25	1.25	1.40	1.36	0.58	0.51	0.62	0.72
	IDW	1.14	1.15	1.40	1.27	0.55	0.43	0.55	0.63
	Attentive LR	1.23	1.26	1.35	1.34	0.58	0.51	0.61	0.70
	k-NN Avg	1.25	1.25	1.40	1.36	0.58	0.51	0.62	0.72
	1-NN	0.97	1.05	1.42	1.18	0.53	0.44	0.62	0.69
ECFP	Nadaraya–Watson	0.99	1.17	1.20	1.22	0.42	0.45	0.43	0.57
	Scaled Dot-Prod.	1.00	1.17	1.20	1.22	0.43	0.45	0.43	0.57
	IDW	0.92	1.12	1.18	1.17	0.42	0.38	0.39	0.54
	k-NN Avg	1.01	1.18	1.21	1.23	0.44	0.46	0.44	0.58
	1-NN	0.68	0.85	1.25	0.98	0.51	0.48	0.57	0.60

E.5 MALT-Delta Component Ablation

This appendix details the component-level ablation for MALT-Delta (Table 17). The tested ablations remove the attention mechanism (`abl_no_attn`), the bilinear δ-head (`abl_no_bilinear`), the pairwise RDKit edit features (`abl_no_edit`), or the anchor-label prior (`abl_no_label_prior`). Results are computed with the GIN encoder over BACE, ESOL, FreeSolv, and their X-splits.

Table 17: MALT-Delta component ablation on the GIN small set. OOD wins and mean OOD change are computed against MALT-Fusion-GIN.

Variant	**OOD wins**	**Mean OOD Δ%**
Full MALT-Delta	6/6	-16.75
`abl_no_attn`	5/6	-12.91
`abl_no_bilinear`	5/6	-15.45
`abl_no_edit`	4/6	-2.06
`abl_no_label_prior`	4/6	-4.67

Shared-neighbor label control. Figure 3 compares three quantities aggregated over all 8 MoleculeNet OOD cells and 5 seeds: MALT-Fusion, MALT-Delta, and a raw top-k label mean on the

MALT-Fusion retrieved set. For GIN and Chemprop, the label mean is much worse than either learned head, so MALT-Delta's gains cannot be explained by direct neighbor-label readout. For UniMol, the matched-initialization runs in Table 2 give 0.506 aggregate OOD MAE for MALT-Delta versus 0.469 for MALT-Fusion, so explicit label conditioning adds no value in this encoder regime.

E.6 Activity Cliffs Summary

We evaluate on the Activity Cliffs benchmark of Van Tilborg et al. (2022), where the OOD test set consists of molecule pairs with high structural similarity but at least tenfold potency difference. Across 30 ChEMBL targets, both MALT heads lower median RMSE relative to the matched inductive baseline for GIN and Chemprop, with the larger gain on GIN. UniMol is already a strong base encoder: both MALT heads improve on Base-UniMol only slightly (median OOD RMSE -0.9% for MALT-Fusion and -1.6% for MALT-Delta), and MALT-Delta-UniMol achieves the most top-2 OOD rankings.

Table 18: Activity Cliffs benchmark summary over 30 ChEMBL targets and 5 seeds. Median RMSE $\Delta\%$ is computed per target as $(\text{method}-\text{base})/\text{base}\times 100$ and then medianed across targets. Negative values indicate improvement. Top-2 OOD counts the number of targets where the method ranks ≤ 2 in OOD RMSE within the 9-method field.

Encoder	Method	ID $\Delta\%$	OOD $\Delta\%$	Top-2 OOD
GIN	Base	—	—	0
	Fusion	-13.4	-14.9	4
	Delta	$\mathbf{-17.0}$	$\mathbf{-15.8}$	**5**
Chemprop	Base	—	—	0
	Fusion	$\mathbf{-9.7}$	$\mathbf{-8.6}$	**4**
	Delta	-8.1	-6.1	3
UniMol	Base	—	—	5
	Fusion	$\mathbf{-2.1}$	-0.9	16
	Delta	$\mathbf{-2.1}$	$\mathbf{-1.6}$	**23**

The full per-target Activity Cliffs tables are omitted from this shortened supplement.

E.7 Mechanism Diagnostics

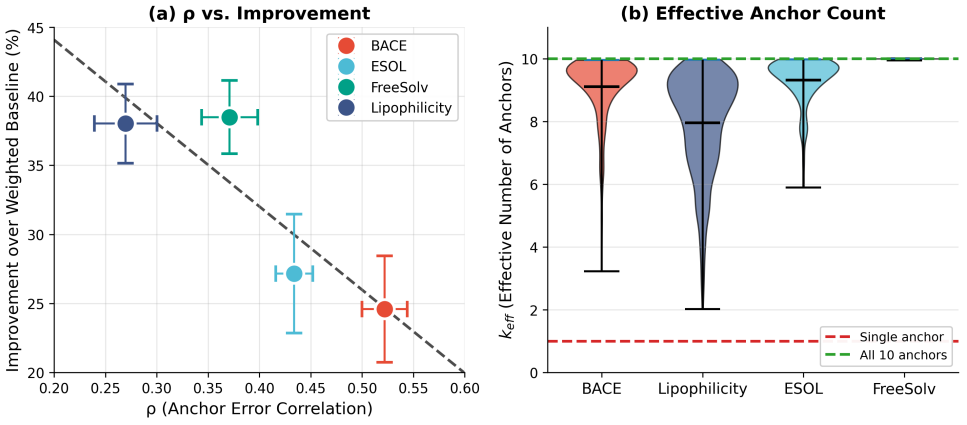
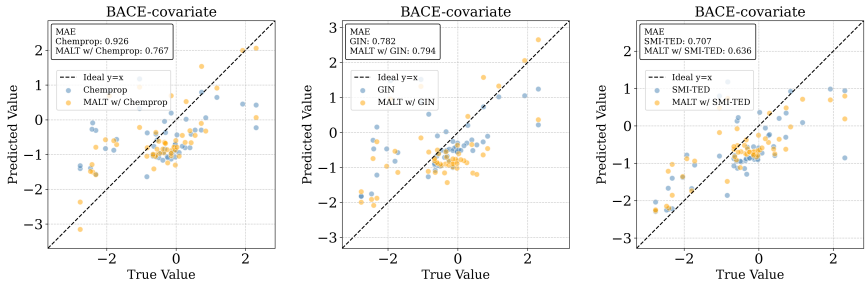


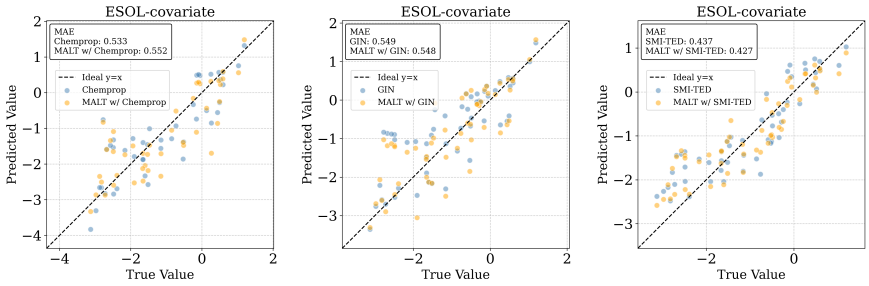
Figure 9: **Multi-anchor mechanism diagnostics.** The MALT benefit is largest when retrieved anchors have less correlated errors, and the learned attention keeps the effective anchor count close to the full retrieved set rather than collapsing to a single nearest neighbor.

E.8 Prediction Parity Plots

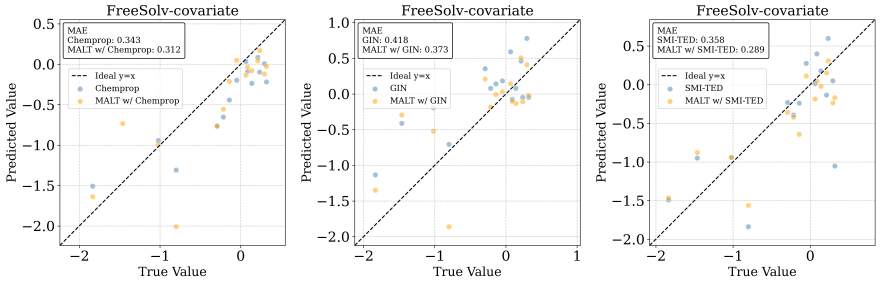
To assess prediction fidelity in both inductive and transductive settings, we show parity plots comparing predicted and true values across datasets and split conditions. Figures 10 and 11 compare Chemprop, GIN, and SMI-TED (Soares et al., 2025) against their MALT-enhanced variants on covariate and label splits. Across base architectures, the MALT variants stay closer to the ideal identity line and achieve lower MAE. Figure 12 compares BLT-RDKit with MALT-Fusion-RDKit across four datasets and two OOD splits; in most cases MALT tightens alignment with the ground truth and reduces the error spread.



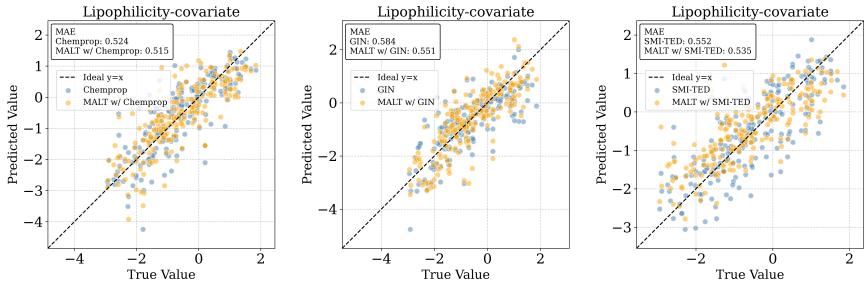
(a) BACE covariate parity plot



(b) ESOL covariate parity plot

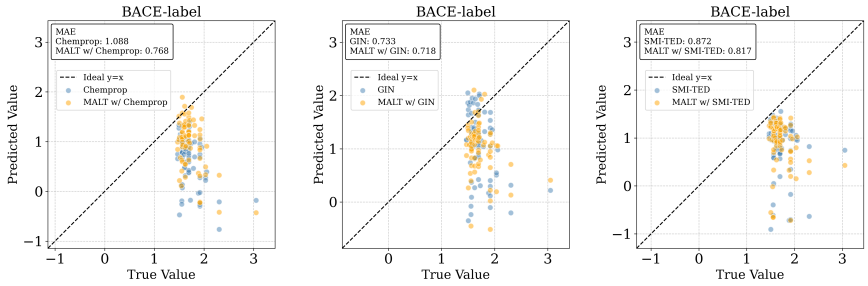


(c) FreeSolv covariate parity plot

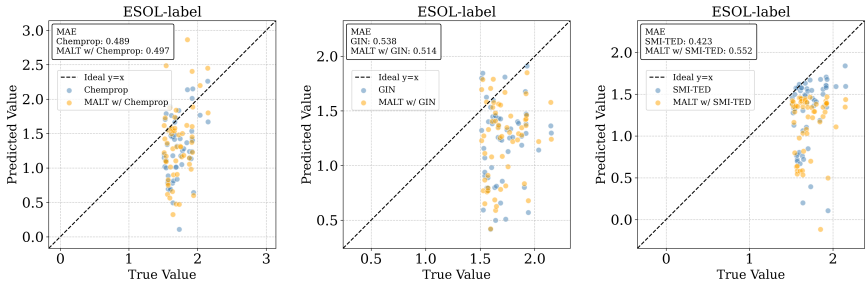


(d) Lipophilicity covariate parity plot

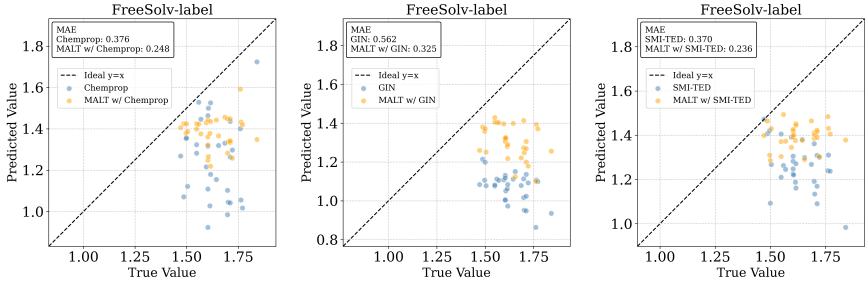
Figure 10: Parity plots comparing inductive models and their MALT-enhanced variants under the covariate split.



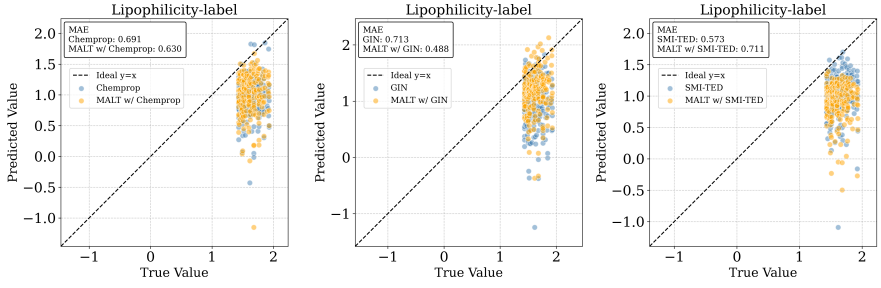
(a) BACE label-split parity plot



(b) ESOL label-split parity plot



(c) FreeSolv label-split parity plot



(d) Lipophilicity label-split parity plot

Figure 11: Parity plots comparing inductive models and their MALT-enhanced variants under the label split.

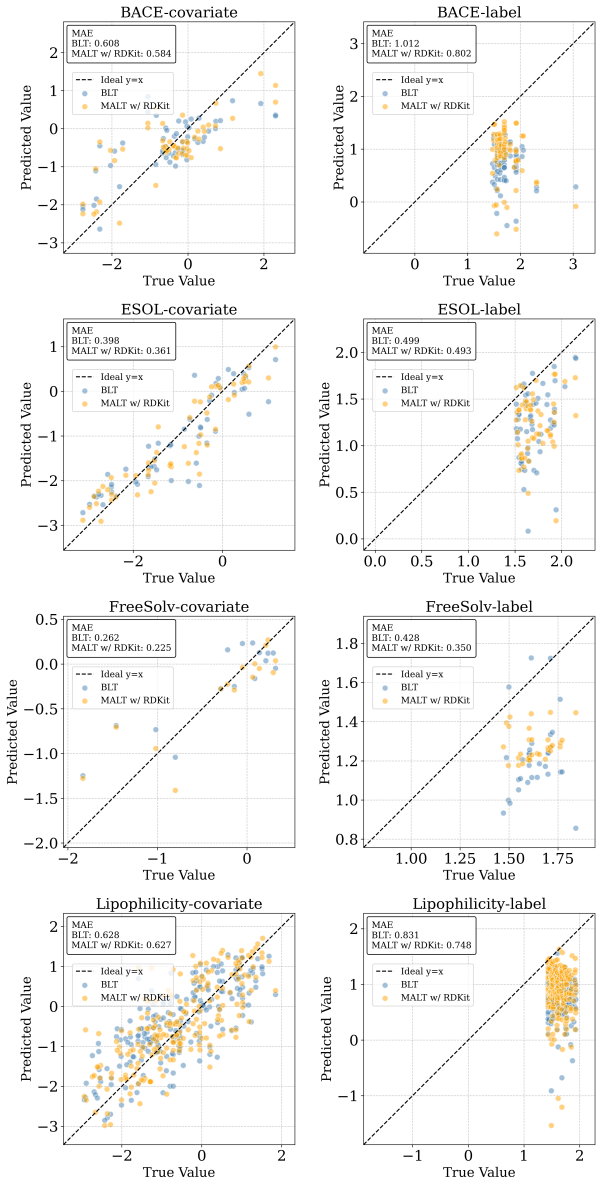
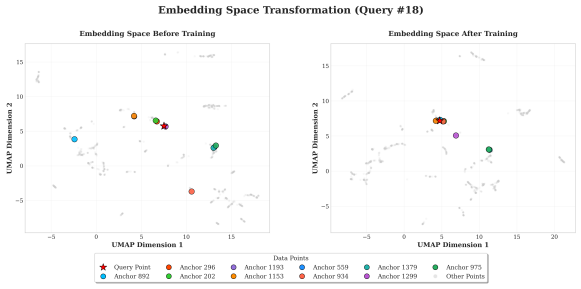


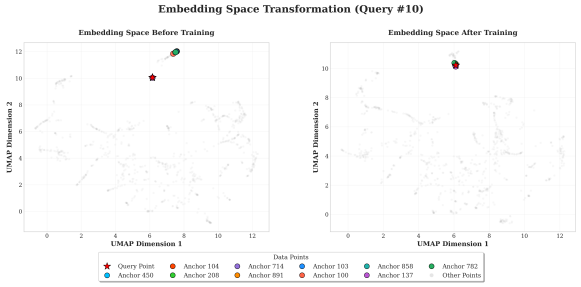
Figure 12: Parity plot comparison between BLT-RDKit and MALT-Fusion-RDKit across datasets and shifts.

E.9 Embedding Space Transformation Comparison

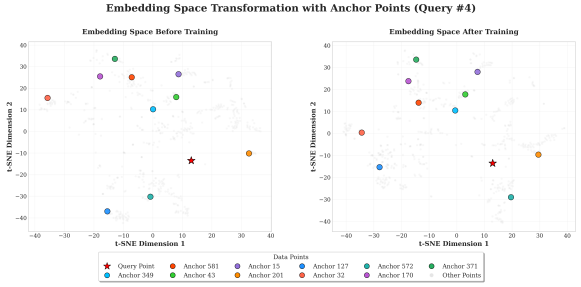
To visualize how MALT reshapes the representation space, we compare the latent space before and after joint training with t-SNE. In Figures 13 and 14, the selected query point (red star) becomes more tightly clustered with its retrieved anchors after training. This contraction appears consistently across datasets and across both covariate and label splits, suggesting that end-to-end training aligns semantically relevant molecules in latent space and improves OOD retrieval quality.



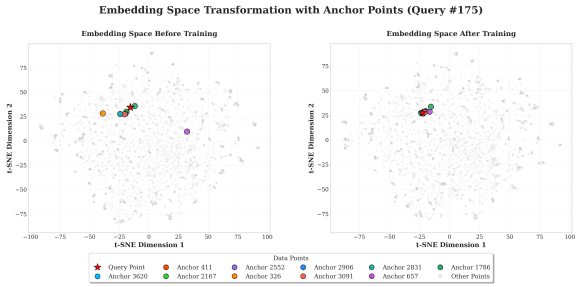
(a) BACE covariate embedding space before vs. after training



(b) ESOL covariate embedding space before vs. after training

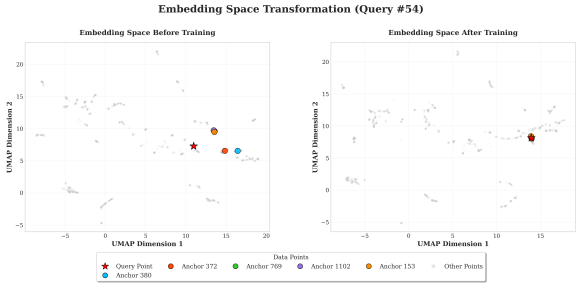


(c) FreeSolv covariate embedding space before vs. after training

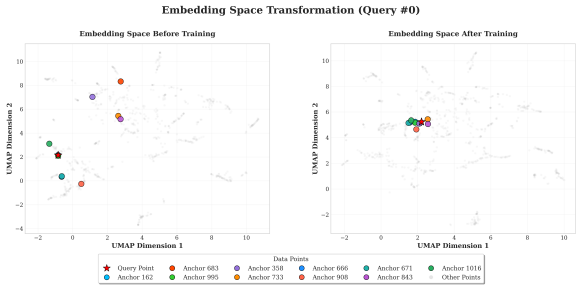


(d) Lipophilicity covariate embedding space before vs. after training

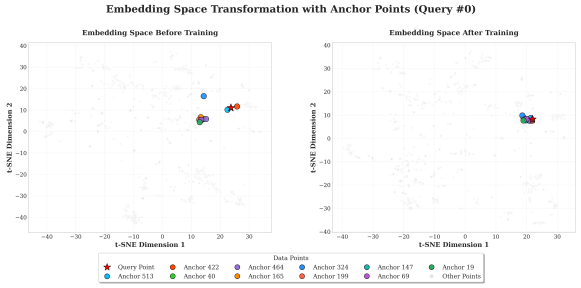
Figure 13: Embedding-space transformation of selected anchors before and after MALT training with GIN under the covariate split.



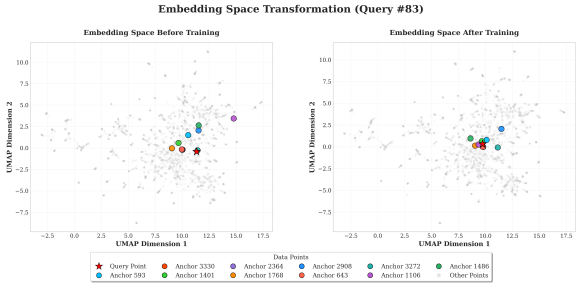
(a) BACE label-split embedding space before vs. after training



(b) ESOL label-split embedding space before vs. after training



(c) FreeSolv label-split embedding space before vs. after training



(d) Lipophilicity label-split embedding space before vs. after training

Figure 14: Embedding-space transformation of selected anchors before and after MALT training with GIN under the label split.

E.10 Robustness Under Harder OOD Regimes

Notation. T1, T2, and T3 are tertiles of structural novelty, measured as Tanimoto distance to the nearest training scaffold; T3 is the most distant and hardest tier. Q1–Q4 are quartiles of the OOD-test property value; Q4 is the most extreme and hardest quartile.

The anchor-distance and OOD-severity visualizations are promoted to Figures 1 and 2 in the main paper.

E.11 Retrieval Quality and Displacement Probe

Retrieval quality. For each seed we fix the trained MALT-Fusion-GIN model, memory bank, queries, and prediction code, and change only the ten anchor indices gathered at inference. Table 19 reports Y-split OOD MAE over five seeds; the increase is the macro-average of per-dataset relative increases. These runs used an H100, so the nearest-anchor row differs slightly from Table 2.

Displacement probe. Molecules (not pairs) are split 60/20/20 into probe-fit, validation, and test groups, and pairs are formed within each group. For $\Delta z = z_i - z_j$ and $\Delta y = y_i - y_j$ we fit a no-intercept ridge probe and report the relative residual $r = \|\Delta y - w^\top \Delta z\|^2 / \|\Delta y\|^2$ (lower is better; r=1 is the zero predictor). "Fixed-pair" evaluates every representation on the same molecule pairs; "own-NN" pairs each test molecule with its nearest neighbors in that representation. Values are averaged over the four MoleculeNet datasets and three replicates (Table 20). The descriptor row uses the normalized 200-dimensional RDKit vector used by BLT.

Table 19: Y-split OOD MAE of MALT-Fusion-GIN when only the test-time anchors change.

Anchors	BACE	ESOL	FSolv	Lipo	Mean	Increase
Nearest 1–10	0.749	0.515	0.317	0.491	0.518	—
Ranks 50–59	0.775	0.528	0.478	0.513	0.574	+15.3%
Random 10	0.769	0.618	0.725	0.517	0.657	+39.3%
Farthest 10	1.328	0.916	1.451	0.620	1.079	+134.7%

Table 20: Displacement probe residual r (lower is better).

Representation	Fixed-pair	Own-NN
Morgan / ECFP4	0.504	0.641
RDKit descriptors	0.291	0.440
Frozen GIN	0.390	0.600
Fine-tuned GIN	0.133	0.565
MALT-Fusion-GIN	**0.083**	**0.263**
MALT-Delta-GIN	0.108	0.446
Random projection	1.000	1.070

F COMPUTATIONAL OVERHEAD ANALYSIS

To assess scalability and practicality, we analyze computational overhead in training, inference, and memory-bank construction.

F.1 TRAINING OVERHEAD: ANCHOR SELECTION AND MEMORY UPDATES

We compare the transductive MALT framework against a standard inductive GNN on BACE and QM9. The primary additional costs are memory update and anchor selection. As shown in Table 21, total overhead per epoch is $+52\%$ on BACE and $+43\%$ on QM9, relative to the inductive model's wall time. For larger datasets, the memory bank can be refreshed every N epochs rather than every epoch; in our reported experiments, we refresh the memory bank every epoch.

Table 21: Per-epoch training time breakdown. Values in each cell are shown for BACE → QM9.

Training Component	Inductive Model (s)	MALT Framework (s)	Overhead (s)
Forward Pass (Total)	0.0693 → 14.9002	0.0800 → 17.7365	+0.0107 → +2.8364
Query Embedding Extraction	0.0000 → 0.0000	0.0633 → 9.0935	+0.0633 → +9.0935
Anchor Selection	0.0000 → 0.0000	0.0052 → 7.0040	+0.0052 → +7.0040
Prediction Head	0.0000 → 0.0000	0.0115 → 1.6390	+0.0115 → +1.6390
Backward Pass	0.1241 → 8.3738	0.1340 → 8.1957	+0.0099 → −0.1781
Memory Update	0.0000 → 0.0000	0.0993 → 12.8468	+0.0993 → +12.8468
Data Loading	0.0065 → 5.6589	0.0042 → 4.2272	−0.0023 → −1.4316
Other Overhead	0.0367 → 4.1498	0.0418 → 4.2201	+0.0051 → +0.0703
Wall Time per Epoch	**0.2366 → 33.0827**	**0.3592 → 47.2264**	**+0.1226 → +14.1437**

F.2 INFERENCE OVERHEAD: BRUTE FORCE VS. FAISS

We benchmark the anchor-selection step using brute-force PyTorch search and Faiss (Douze et al., 2025) over memory banks ranging from 10k to 10M vectors and embedding dimensions 256, 128, and 64. Faiss is competitive at large bank sizes ($\geq$ 500k) with embedding dimension ≥ 256; at smaller scales, PyTorch brute-force search is comparable or faster.

Table 22: Faiss vs. PyTorch nearest-neighbor search. Values in each cell are shown for embedding dimensions $D = 256 \rightarrow D = 128 \rightarrow D = 64$.

Bank Size	PyTorch Init (ms)	Faiss Init (ms)	PyTorch Search (ms)	Faiss Search (ms)	Speedup
10k	1.84 → 1.89 → 1.70	91.5 → 160 → 93.1	0.22 → 0.16 → 0.15	0.32 → 0.31 → 0.36	0.68× → 0.50× → 0.41×
100k	16.9 → 20.2 → 12.4	162 → 228 → 116.5	1.45 → 1.00 → 0.62	1.47 → 1.76 → 2.26	0.99× → 0.57× → 0.27×
500k	128.7 → 86.3 → 94.5	394 → 306 → 212	7.24 → 5.09 → 3.09	6.57 → 7.79 → 10.28	**1.10**× → 0.65× → 0.30×
1M	175.1 → 148.2 → 129.6	672 → 506 → 382	14.48 → 10.20 → 6.14	12.94 → 15.44 → 20.39	**1.12**× → 0.66× → 0.30×
10M	2706.7 → 2240.5 → 1308.4	6317 → 4652 → 2653	134.70 → 102.20 → 61.44	127.70 → 152.00 → 202.10	**1.05**× → 0.67× → 0.30×

F.3 MEMORY BANK CONSTRUCTION: BUILD TIME AND MEMORY FOOTPRINT

We evaluate the one-time cost of building the memory bank on datasets ranging from 10^3 to 10^6 molecules. This procedure is performed once with gradients disabled to extract and store each molecule's embedding. Table 23 shows that the process scales approximately linearly with dataset size, with embedding extraction as the main bottleneck.

Table 23: Memory-bank construction time and size scaling.

Dataset Size	Total Time (s)	Data Loading (s)	Embedding Extraction (s)	Concatenation (s)	Memory Size (MB)
1,000	0.274	0.004	0.194	0.028	0.98
5,000	0.771	0.006	0.540	0.014	4.88
10,000	1.594	0.007	1.106	0.018	9.77
50,000	7.709	0.044	5.517	0.148	48.83
100,000	15.538	0.096	10.872	0.321	97.66
500,000	75.837	0.396	53.686	1.782	488.28
1,000,000	151.713	0.790	108.333	3.147	976.56