ACTIVE MOLECULAR QUERY SYNTHESIS BY INFLUENCE-GUIDED LATENT SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Molecular screening relies on accurate property predictions, but obtaining the data to train these predictors can require costly assays or computations. Active learning reduces this cost by selecting molecules for evaluation, but its choices are often restricted to an existing library. Molecular generation allows the learner to construct additional queries and choose them for their estimated effect on prediction. We propose Influence-guided Molecular Query Synthesis (iMQS) to computationally generate molecular structures for their value as training data under a limited evaluation budget. Its acquisition objective is the estimated reduction in prediction loss on a validation set. An influence function approximates this improvement without retraining for every candidate and guides search in a continuous molecular representation. Our analysis bounds errors in comparing candidate improvements and the validation improvement lost through query selection when the predictor's final layer is refitted. Across nine molecular endpoints, iMQS improves predictive learning over several pool and generation baselines. In two AutoDock Vina applications, mean recovery of measured screening actives increases over the acquisition and refitting procedure. These results support molecular query generation as a way to acquire useful training data under limited evaluation budgets.

1 INTRODUCTION

Molecular property prediction helps prioritize compounds for experimental evaluation and is an important part of data-driven drug discovery (Wu et al., 2018). Its accuracy depends on labeled molecular data, which can be expensive to obtain through laboratory assays or detailed computational evaluations (Eckmann et al., 2025). Active learning addresses this cost by choosing which molecules to evaluate next. In molecular applications, it directs evaluation toward a small part of a large library, using the observations to improve subsequent predictions and selections (Graff et al., 2021; van Tilborg & Grisoni, 2024). The objective is to build a predictor within a limited evaluation budget.

A fixed candidate library limits which compounds can be evaluated. Molecular generative models expand this choice by constructing additional structures (Gómez-Bombarelli et al., 2018). Property-driven generation searches for compounds with favorable predicted properties, such as strong binding to a target (Eckmann et al., 2022). Yet the predicted property of a compound and the improvement obtained by adding its observed property to the training set are different quantities. Active learning can improve molecular predictors by selecting generated candidates for evaluation (Nahal et al., 2024). The resulting active learning problem is to construct a query whose observed label improves predictions for other molecules. This gives generation a direct criterion for expanding the training set.

We evaluate useful training data by the prediction improvement they produce on a separate set of labeled reference molecules. These molecules specify where accurate predictions are needed. This choice follows work that selects training data for validation performance (Killamsetty et al., 2021) and directs active learning toward predictions on relevant inputs (Bickford Smith et al., 2023). Influence-guided Molecular Query Synthesis (iMQS) uses the estimated reduction in reference prediction loss to construct molecular candidates. An influence function approximates the effect of adding an observation, avoiding a separate model fit for every candidate. We optimize this estimate in a latent space, a continuous vector representation that a molecular generator converts into structures. The evaluated structures and their observed properties expand the training set; refitting the predictor then updates the criterion for the next query. The theory connects query selection to predictive

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.

improvement. With the predictor's earlier layers fixed, we bound the error in comparing validation-loss reductions after its final layer is refitted, and the improvement lost relative to the best candidate in a finite set. The bound accounts for incomplete fitting, numerical error, and disagreement between the label used during search and the observed outcome. Section 4 states the result.

We evaluate iMQS in two complementary settings. Across nine molecular endpoints labeled by a fixed TxGemma teacher, we compare query synthesis with pool selection and other generation methods under limited acquisition budgets. iMQS reduces mean test error from initialization on all nine endpoints, with recurring gains over generation baselines and more variable differences from the strongest pool methods. We then use AutoDock Vina labels to train predictors for two screening tasks, evaluating their recovery of experimentally measured actives in separate LIT-PCBA bioassay libraries (Tran-Nguyen et al., 2020). For iMQS, mean average precision increases over acquisition and refitting. Together, these results support generating molecular queries to expand useful training data, with gains that depend on the endpoint and comparison method.

2 Related work

Active learning and query synthesis. Active learning selects examples for their expected value to prediction (Bickford Smith et al., 2023; van Tilborg & Grisoni, 2024). Pool-based criteria include predictive uncertainty and coverage of the available data (Ash et al., 2020; Sener & Savarese, 2018). Predictive criteria estimate how new labels would reduce future error (Roy & McCallum, 2001), improve validation performance (Killamsetty et al., 2021), or reduce uncertainty in predictions on relevant inputs (Bickford Smith et al., 2023). Influence-based criteria approximate the effect of model updates on prediction loss (Liu et al., 2021; Yao et al., 2026). Query synthesis extends acquisition beyond a fixed pool by constructing inputs (Angluin, 1988; Zhu & Bento, 2017; Schumann & Rehbein, 2019). For molecules, the optimized representation must yield a structure that the evaluator can assess. iMQS connects these steps by using the prediction-loss to guide latent search before decoding.

Molecular generation for properties and learning. Learned molecular representations support continuous search for desired properties (Gómez-Bombarelli et al., 2018; Krenn et al., 2020). Latent inceptionism for targeted molecule generation (LIMO) and constrained latent optimization direct this search using property predictions (Eckmann et al., 2022; Griffiths & Hernández-Lobato, 2020). Feedback from new evaluations can then update generation or candidate selection (Kyro et al., 2024; Eckmann et al., 2025). Closest to our objective, Nahal et al. (2024) apply expected predictive information gain (EPIG), which measures information about predictions on reference inputs, to candidates from a molecular generator. iMQS directly optimizes an influence estimate of learning value with respect to the latent features.

Influence and finite model updates. Influence functions approximate model changes through local derivatives (Koh & Liang, 2017). Finite reweighting analyses control the error beyond infinitesimal changes (Giordano et al., 2019; Koh et al., 2019); Hessian-metric and self-concordant analyses retain local curvature information (Fisher et al., 2023; Rubinstein & Hopkins, 2026). For deep models, interpretation also depends on the parameters refitted and the optimization procedure (Basu et al., 2021; Bae et al., 2022). Shared effects of incomplete fitting appear in Koh & Liang (2017, Appendix B), and Wei et al. (2025) subtract the effect of further training on the unchanged dataset. For query selection, the approximation must preserve the ordering of candidates (Hu et al., 2024). We compare two finite head refits directly, retaining differences in their gradients, curvature, and fitting residuals. Section 4 explains how this comparison yields the query-ranking bound.

3 iMQS: generating molecules for learning

At each cycle, iMQS constructs a molecular query to improve the predictor within the remaining labeling budget. We score a proposed training example by its estimated reduction in reference prediction loss, search for a high-scoring molecular representation, and decode it into a structure. The evaluator, called the oracle, returns the property used to update the predictor. It is a docking calculation in Section 5.3 and a fixed predictive model, called a teacher, in Section 5. The same procedure handles discrete class labels and numerical responses. Classification uses cross-entropy loss; regression uses squared-error loss, with derivatives in Section A.2.

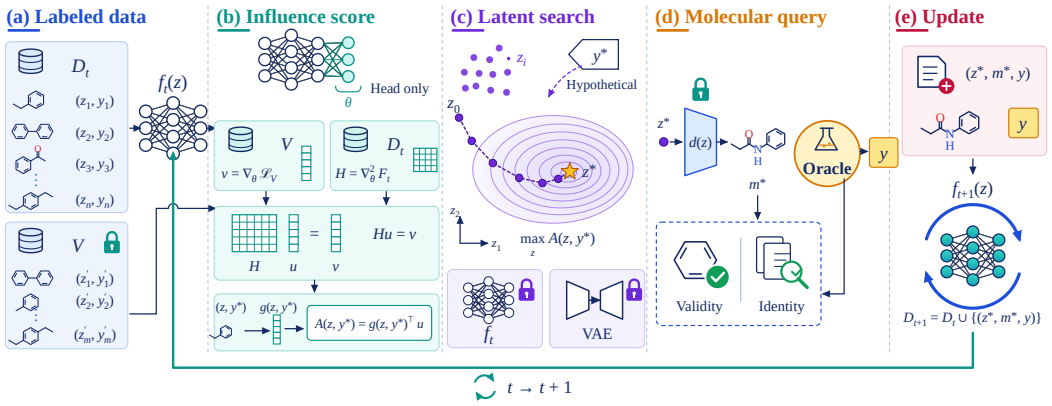


Figure 1: iMQS acquisition cycle. An influence score guides latent search; a valid decoded molecule is evaluated, and its observed label updates the training set.

3.1 Preliminaries

We optimize continuous features, but the oracle evaluates molecules. To connect these two spaces, let $\mathcal{M}$ be the molecule space and o a callable property evaluator. For classification, a probability score $o(m) \in [0, 1]$ is thresholded at a fixed τ to obtain $y(m) = \mathbf{1}\{o(m) \geq \tau\}$; the multiclass notation uses $\mathcal{Y} = \{0, \ldots, C-1\}$. For regression, $y(m)$ is the numerical response, standardized using the mean and standard deviation of the initial training labels. These normalization constants remain fixed during acquisition. A pretrained variational autoencoder (VAE) maps molecular structures to continuous feature vectors through an encoder and maps these vectors back to structures through a decoder. Both remain fixed throughout acquisition:

$$\mathrm{Enc} : \mathcal{M} \to \mathbb{R}^p, \qquad \mathrm{Dec} : \mathbb{R}^p \to \mathcal{M} \cup \{\perp\}, \tag{1}$$

where $\perp$ denotes an invalid output. We use the mean of the encoder's distribution as the molecular feature vector and a fixed token-selection rule to decode each vector.

Each labeled record retains its feature, molecule, and observed label. At round t, write $\mathcal{D}_t = \{(z_i, m_i, y_i)\}_{i=1}^{n_t}$. An initial molecule has $z_i = \mathrm{Enc}(m_i)$, whereas a generated record stores its optimized feature z_i, decoded molecule $m_i = \mathrm{Dec}(z_i)$, and observed label. A reference set $\mathcal{V}$ and test set $\mathcal{T}$ are randomly selected during data splitting and remain fixed throughout each run. Their molecules are represented by the fixed encoder's mean features.

The predictor has a multilayer perceptron (MLP) body ϕ_ω and a linear prediction head h_θ. For classification, $f_{\omega,\theta}(z) = \mathrm{softmax}(h_\theta(\phi_\omega(z)))$ returns class probabilities and $\ell(\omega, \theta; z, y) = -\log f_{\omega,\theta}(z)_y$. For regression, $f_{\omega,\theta}(z) = h_\theta(\phi_\omega(z))$ is scalar and $\ell(\omega, \theta; z, y) = \frac{1}{2}(f_{\omega,\theta}(z) - y)^2$. The mean training loss is

$$L_t(\omega, \theta) = \frac{1}{n_t} \sum_{i=1}^{n_t} \ell(\omega, \theta; z_i, y_i). \tag{2}$$

The full predictor is refitted on the current labeled set at each acquisition cycle by minimizing $L_t + \lambda \|(\omega, \theta)\|^2/2$, where $\lambda > 0$ penalizes large parameter values. Section A.3 gives the fitting convention.

3.2 The acquisition score

A useful new observation should improve predictions on molecules beyond the acquired training set. We therefore measure learning value by the reduction in average prediction loss on the labeled reference set $\mathcal{V}$, following validation-based data selection (Killamsetty et al., 2021). We use "validation loss" to refer to this reference-set loss. For a fixed predictor, loss on independently sampled reference observations estimates expected loss on new observations from the same distribution. The reference set guides acquisition; a separate test set evaluates the resulting predictor. Section A.4 explains this

distinction. The reference loss is

$$\mathcal{L}_{\mathcal{V}}(\omega, \theta) = \frac{1}{|\mathcal{V}|} \sum_{(m,y)\in\mathcal{V}} \ell(\omega, \theta; \mathrm{Enc}(m), y), \qquad V_t(\theta) = \mathcal{L}_{\mathcal{V}}(\omega_t, \theta). \tag{3}$$

Here V_t fixes the current predictor body. For a requested label y^*, the ideal query minimizes validation loss after adding (z, y^*) and refitting the predictor:

$$\min_{z\in\mathbb{R}^p} \mathcal{L}_{\mathcal{V}}\big(\omega_t^+(z, y^*), \theta_t^+(z, y^*)\big), \tag{4}$$

$$\text{where } (\omega_t^+, \theta_t^+)(z, y^*) \in \arg\min_{\omega,\theta} \left\{ \frac{n_t L_t(\omega, \theta) + \ell(\omega, \theta; z, y^*)}{n_t + 1} + \frac{\lambda}{2} \|(\omega, \theta)\|^2 \right\}.$$

The outer search requires repeated model fitting, making direct optimization costly. Influence functions approximate the refit from derivatives at the current predictor. Since the label is unavailable before evaluation, we first choose a requested label and derive the score conditional on it.

3.2.1 Choosing a requested label

The hypothetical update in equation 4 requires a label before the oracle evaluates the molecule. We choose this requested label to direct search toward outcomes that are less represented in the current training set. For classification, it is the least represented class, with cyclic tie-breaking. For regression, we apply the same principle by grouping standardized training responses around three fixed values below, at, and above the initial mean. The least represented group then supplies its fixed value as the requested response. Section A.3 gives the assignment rule.

With the requested label y^* fixed for the current cycle, the influence score below guides molecular search. The oracle subsequently supplies the actual training label, and we retain every valid response, including those that differ from the request. Section 4 accounts for this disagreement when relating the search score to the improvement obtained with the observed label.

3.2.2 Approximating the refit with an influence function

To score many candidates efficiently, we approximate the effect of adding (z, y^*) by updating the final prediction layer while holding the earlier layers ω_t fixed. The regularized head objective is $F_t(\theta) = L_t(\omega_t, \theta) + \lambda \|\theta\|^2 / 2$. Define the candidate gradient, training Hessian, and validation gradient at the current checkpoint by

$$g_t(z, y) = \nabla_\theta \ell(\omega_t, \theta_t; z, y), \qquad \widehat{H}_t = \nabla_\theta^2 F_t(\theta_t), \qquad v_t = \nabla_\theta V_t(\theta_t). \tag{5}$$

Training data determine the curvature $\widehat{H}_t$, and reference data determine the loss gradient v_t. The positive regularizer makes the head Hessian invertible. Let $\widehat{u}_t$ be the numerical solution of $\widehat{H}_t \widehat{u}_t = v_t$, and let $\Delta_t(z, y^*)$ denote the validation-loss reduction after adding (z, y^*) and refitting the head. The standard influence approximation at a stationary head (Koh & Liang, 2017) motivates the computed score

$$\Delta_t(z, y^*) \approx \frac{A_t(z, y^*) - \kappa_t}{n_t + 1}, \qquad A_t(z, y^*) = g_t(z, y^*)^\top \widehat{u}_t \approx g_t(z, y^*)^\top \widehat{H}_t^{-1} v_t. \tag{6}$$

For an exact solve, the computed score equals the inverse-Hessian expression. One solve supplies $\widehat{u}_t$ for every candidate in the cycle. The offset $\kappa_t = \widehat{u}_t^\top \nabla_\theta L_t(\omega_t, \theta_t)$ accounts for rescaling the existing records when one example is added. Since the offset and scale are shared by all candidates, maximizing A_t maximizes the predicted loss reduction. Section A gives the derivative calculation; Section 4 bounds the finite-refit and numerical errors at the implemented checkpoint.

Gradient ascent on A_t guides the latent search. The predictor, training and reference data, and $\widehat{u}_t$ remain fixed during this search, so candidate optimization avoids repeated model fitting. Section A.3 gives the derivative.

3.3 From the acquisition score to a molecular query

Labeled features provide starting points for query construction. We draw z_0 uniformly from the stored features in $\mathcal{D}_t$ and optimize $\max_{\delta\in\mathbb{R}^p} A_t(z_0 + \delta, y_t^*)$. Starting at $\delta = 0$, Adam follows equation 16

Algorithm 1 Influence-guided Molecular Query Synthesis (iMQS)

Require: Fixed encoder and decoder; labeled $\mathcal{D}_0$; reference $\mathcal{V}$; oracle; acquisition budget
Set $t = 0$; initialize identity exclusions and oracle ledger; fit the predictor on $\mathcal{D}_0$
while acquisition budget remains **do**
  Choose requested label y_t^* from the least represented class or response group
  Freeze the predictor; solve $\widehat{H}_t \widehat{u}_t = v_t$; set $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t$
  repeat ▷ Keep y_t^* fixed throughout failed-call retries
    repeat
      Draw z_0 uniformly from stored features; optimize $A_t(z, y_t^*)$ with fresh randomness
      Decode the final iterate $m^* = \mathrm{Dec}(z^*)$
    until m^* is valid and its identity is eligible
    Query the oracle; update the identity ledger and the declared acquisition budget
    if a label y is returned **then**
      Add (z^*, m^*, y) to $\mathcal{D}_{t+1}$, including when $y \neq y_t^*$
    end if
  until a unit of the declared acquisition budget is consumed
  Reset and refit the predictor on $\mathcal{D}_{t+1}$; advance t
end while
return predictor and labeled data

for a fixed number of steps. Every latent coordinate is free to change, while the predictor and VAE remain fixed. The final iterate z^* defines the query proposal.

To obtain an evaluable structure, we decode $m^* = \mathrm{Dec}(z^*)$ and check its validity and identity before oracle access. Invalid or excluded structures trigger a fresh independent seed and random-number stream within the same cycle. Once an eligible molecule is found, we submit it to the oracle. Under a paid-attempt budget, the call is charged regardless of its outcome. Every returned label is retained, including when $y(m^*) \neq y_t^*$: we add $(z^*, m^*, y(m^*))$ to $\mathcal{D}_{t+1}$. The full predictor is then refitted on the augmented set. Retaining z^* trains on the feature scored during search, whereas molecules in the reference and test sets use encoder features. Decoding and re-encoding need not return the same vector. The observed label links the generated structure to the training update, and held-out prediction evaluates the value of that update. The new record can also initialize later queries. Figure 1 and algorithm 1 summarize the proposal loop; Section C.5 gives the bounded search used for docking.

4 Theoretical analysis

We compare validation gains from finite refits of the regularized prediction head while keeping the earlier layers fixed. Fix a cycle with $n \geq 1$ training records and a nonempty finite set $\mathcal{C}$ of eligible candidates $q = (z_q, y_q)$, where y_q is the oracle label of the decoded molecule. Write $S(z_q) = A_t(z_q, y_t^*)$ for the computed score and $\Delta_q = V_t(\theta_t) - V_t(\theta_q)$ for the validation gain after a possibly inexact head refit θ_q. Assumption B.1 requires smooth losses, uniformly positive head curvature, and bounded curvature changes and refit distances on a common convex region.

Theorem 4.1. *Under Assumption B.1, let $\mathcal{B}_{qr} \geq 0$ be the explicit error bound in equation 23. For any returned candidate $\widehat{q} \in \mathcal{C}$,*

$$\max_{q \in \mathcal{C}} \Delta_q - \Delta_{\widehat{q}} \leq \max_{q \in \mathcal{C}} \left\{ \frac{S(z_q) - S(z_{\widehat{q}})}{n+1} + \mathcal{B}_{q\widehat{q}} \right\}. \tag{7}$$

In particular, if $\widehat{q} \in \arg\max_{q \in \mathcal{C}} S(z_q)$, the right-hand side is at most $\max_{q \in \mathcal{C}} \mathcal{B}_{q\widehat{q}}$.

Interpretation. The left-hand side is the validation improvement lost by choosing $\widehat{q}$ instead of the best candidate in $\mathcal{C}$. The right-hand side separates the score gap left by search from the error in predicting finite-refit gains. If search finds the highest-scoring candidate, only the latter remains: small errors for every candidate mean that the selected query achieves nearly the best validation improvement in the candidate set. The error accounts for finite refits, unequal fitting residuals, numerical solve error, and disagreement between requested and observed labels. Theorem B.2 gives the pairwise bound, explicit terms, and proof. This guarantee concerns fixed-body head refits; full-predictor test performance is assessed empirically.

5 EXPERIMENTS

We compare molecular query synthesis with pool selection and other generation methods on nine molecular endpoints, measuring prediction error and paired differences under limited evaluation budgets. Two docking applications then assess whether the trained predictors recover measured actives from separate screening libraries.

5.1 EXPERIMENTAL SETTINGS

Tasks and oracle. The benchmarks contain four binary endpoints, CYP1A2 Veith, Pgp Broccatelli, Tox21 SR-MMP, and CYP3A4 Veith, and five regression endpoints, KIBA GSK3B, KIBA PRKACA, KIBA VEGFR2, LD50 Zhu, and Lipophilicity AstraZeneca (Huang et al., 2021; Tang et al., 2014). Evaluating newly generated molecules requires a callable *in silico* oracle because a fixed dataset labels only its recorded structures. We use TxGemma-9B-Predict (Wang et al., 2025), built on pretrained Gemma 2 and fine-tuned on therapeutic data, with reported performance competitive with state-of-the-art therapeutic predictors. Its coverage of drug properties and molecular interactions provides a common oracle across the selected endpoints. Holding this teacher fixed gives pool selection and generation the same label source.

Protocol and metrics. Each task uses ten shared random splits, with 100 initial and 100 reference labels. A fixed LIMO VAE supplies molecular features, and an MLP is refitted at each acquisition cycle. We report test negative log-likelihood (NLL) for classification and mean absolute error (MAE) on the normalized teacher-response scale for regression; lower is better. Both training and test labels come from TxGemma, so these metrics measure agreement with the teacher. We also report final-budget accuracy and mean squared error (MSE) to assess class decisions and squared prediction errors. Curves cover 0–150 acquisitions beyond initialization: binary tasks count paid attempts; regression tasks count successfully acquired labels. Lines show means and bands show one sample standard deviation. Section C gives data roles, accounting conventions, and reproducibility information.

Compared methods. Pool methods comprise random selection, entropy sampling (prediction variance for regression), Batch Active Learning by Diverse Gradient Embeddings (BADGE) (Ash et al., 2020), matched pool influence (Pool IF), and Goal-Oriented Influence-Maximizing Data Acquisition (GOIMDA) (Yao et al., 2026). Generation methods comprise random-direction search with a matched search radius (Random-radius), generated-candidate EPIG, the REINVENT/random-forest EPIG method of Nahal et al. (2024) (Nahal-EPIG), property-guided latent search, and iMQS. Pool IF applies the same learning criterion within a fixed library. Random-radius and property-guided search use the same generator and training-feature convention as iMQS, allowing us to compare acquisition objectives. All ten methods have ten splits on every endpoint. For KIBA and Lipophilicity,

Table 1: Paired prediction-error comparisons with iMQS (two-sided t-tests, unadjusted $\alpha = 0.05$, ten splits). Each triplet gives budgets 50/100/150: bold $+$ is a win, 0 is nonsignificant, and $-$ is a loss. W/NS/L counts these outcomes across nine endpoints. *Higher-response configuration on KIBA and Lipophilicity; fixed configuration elsewhere.

	Paired decisions at budgets 50 / 100 / 150									W/NS/L		
Comparator	CYP 1A2	Pgp	SR-MMP	CYP 3A4	GSK3B	PRKACA	VEGFR2	LD50	Lipo.	50	100	150
Random pool	000	000	+++	+00	000	000	0++	+++	000	3/6/0	3/6/0	3/6/0
Entropy / variance	000	000	000	++0	000	0++	+++	+++	0+0	3/6/0	5/4/0	3/6/0
BADGE	000	000	+++	000	+00	0++	+++	+++	0+0	4/5/0	5/4/0	4/5/0
Pool IF	+00	000	+++	+++	++0	0++	+00	+++	000	6/3/0	5/4/0	4/5/0
GOIMDA	+++	++0	+++	+++	000	000	+00	+++	000	6/3/0	5/4/0	4/5/0
Random-radius	00+	+++	0++	000	000	0++	+++	+++	+++	4/5/0	6/3/0	7/2/0
Generated EPIG	+++	+++	+++	000	000	0+0	+++	+++	0++	5/4/0	7/2/0	6/3/0
Nahal-EPIG*	+++	+++	+0+	+++	+++	0++	+++	+++	0+0	7/2/0	8/1/0	8/1/0
Property-guided*	+++	00+	0++	0+0	000	000	00+	0++	+++	2/7/0	5/4/0	6/3/0

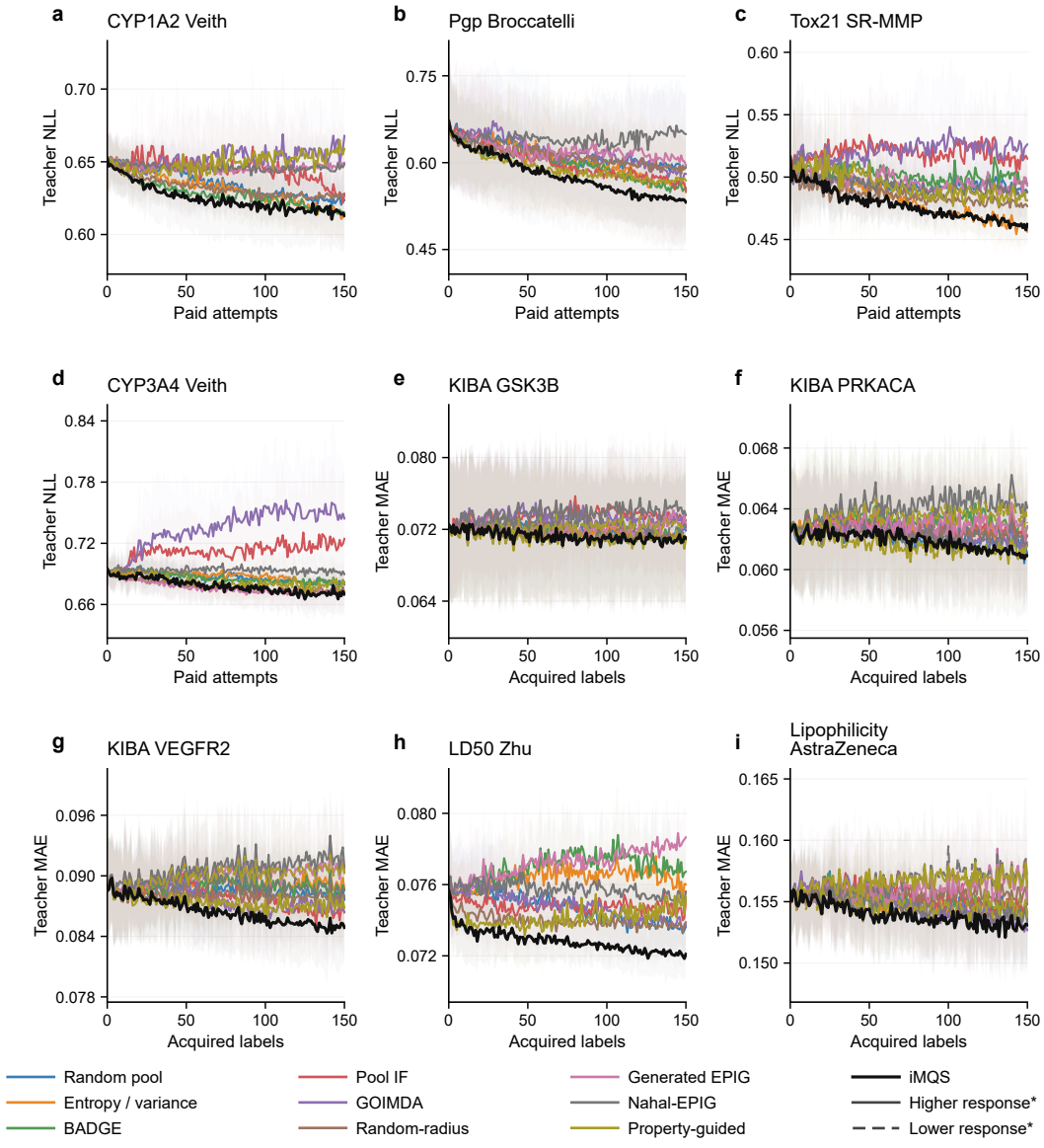


Figure 2: Predictive learning on nine endpoints (mean and sample SD, ten splits). Lower teacher NLL (a–d) or normalized MAE (e–i) is better. *Solid/dashed lines indicate higher/lower-response configurations on KIBA and Lipophilicity.

property-guided search and Nahal-EPIG use both higher- and lower-response configurations; the other methods share one trajectory across these comparisons. Sections C.2 and C.3 give the adaptations and memberships.

5.2 Results of Performance Comparisons

We report the learning curves of compared methods in Figure 2. The performances are the mean of ten random seeds, the shaded area is the standard deviation. A significance test based on significance level $\alpha = 0.05$ is reported in Table 1. Results of more performance metrics are presented in Table 2. From these results, we mainly have the following findings:

Directing generation toward learning improves acquisition. By the final budget, iMQS has lower mean test error than its initial predictor on all nine endpoints. Its advantage over Random-radius

Table 2: Teacher test accuracy and MSE at budget 150. Values are means, with sample SD below in parentheses, over ten splits. Bold/underline mark the best/second-best unrounded means. Both panels use the comparator configurations of Table 1.

(a) Classification: accuracy (%, ↑)

Method	CYP 1A2	Pgp	SR-MMP	CYP 3A4
Random pool	64.3 (1.7)	72.8 (5.0)	78.1 (2.4)	57.2 (2.9)
Entropy pool	66.2 (1.4)	**76.0** (2.7)	**81.0** (0.6)	58.4 (2.1)
BADGE	64.9 (2.3)	74.8 (4.3)	78.4 (1.5)	56.4 (3.6)
Pool IF	<u>66.4</u> (1.2)	74.3 (3.9)	75.1 (2.3)	58.8 (5.4)
GOIMDA	66.0 (3.6)	<u>75.6</u> (2.6)	79.5 (1.8)	**60.7** (1.5)
Random-radius	64.2 (1.9)	72.1 (4.2)	79.3 (2.3)	58.4 (2.0)
Generated EPIG	60.9 (2.6)	69.3 (3.8)	<u>79.5</u> (2.0)	58.9 (2.1)
Nahal-EPIG	61.7 (2.5)	67.4 (5.8)	78.5 (1.8)	54.6 (3.8)
Property-guided	61.5 (3.3)	73.3 (3.4)	78.0 (2.0)	56.1 (3.7)
iMQS	**66.7** (2.4)	75.2 (3.2)	79.3 (1.8)	<u>59.9</u> (4.0)

(b) Regression: MSE ($\times 10^{-3}$, ↓)

Method	KIBA GSK3B	KIBA PRKACA	KIBA VEGFR2	LD50	Lipo.
Random pool	9.44 (1.66)	<u>6.56</u> (0.69)	12.19 (0.80)	9.40 (0.49)	<u>34.56</u> (1.65)
Variance pool	<u>9.23</u> (1.78)	6.66 (0.84)	12.02 (0.75)	9.81 (0.73)	34.81 (1.78)
BADGE	9.34 (1.83)	6.74 (0.79)	<u>11.96</u> (0.77)	10.04 (0.64)	34.94 (1.96)
Pool IF	9.61 (1.72)	6.71 (0.82)	11.99 (0.77)	9.67 (0.71)	35.79 (1.55)
GOIMDA	9.34 (1.43)	6.68 (0.57)	12.16 (0.73)	<u>9.31</u> (0.51)	35.75 (2.30)
Random-radius	9.48 (2.03)	6.80 (0.90)	12.28 (1.03)	9.50 (0.46)	35.00 (1.69)
Generated EPIG	9.54 (1.99)	6.90 (0.83)	13.02 (1.03)	10.35 (0.62)	34.69 (1.45)
Nahal-EPIG	9.52 (1.94)	6.87 (0.74)	12.50 (0.98)	10.01 (0.47)	34.70 (1.64)
Property-guided	9.44 (1.95)	6.84 (0.91)	12.31 (1.07)	9.64 (0.64)	35.11 (1.70)
iMQS	**9.21** (1.77)	**6.54** (0.83)	**11.47** (0.94)	**9.11** (0.47)	**34.38** (1.76)

supports using the predicted learning gain to guide search within the shared generator. Gains over generated-candidate EPIG are also common. Thus, the expanded query space is useful when the search criterion identifies observations that improve the predictor. Comparisons with Pool IF evaluate the complete synthesis procedure, which changes both the candidate molecules and their training features (Section C.2).

Gains depend on the endpoint and budget. The clearest recurring gains over the baselines occur on LD50, SR-MMP, and VEGFR2, whereas the strongest pool methods remain competitive on several other endpoints. We compare matched splits at budgets 50, 100, and 150 using exploratory, unadjusted two-sided paired t-tests at $\alpha = 0.05$. Table 1 presents all nine endpoints and three budgets in one table. Significant gains are common against baselines, while comparisons with the leading pool methods often remain nonsignificant, including cases where their final mean error is lower. No significant loss is observed in these pointwise tests. Section C.4 gives the pairing and test details.

Learning-directed search often improves on property-guided generation. The comparison with property-guided generation holds the VAE and latent optimizer fixed while changing the acquisition objective. Its predictive error is often higher than that of iMQS, and paired gains for iMQS are more common at later budgets. The higher- and lower-response configurations nevertheless remain competitive on some endpoints, including GSK3B, PRKACA, and Lipophilicity. These results support choosing queries by their estimated contribution to prediction, with the benefit depending on the endpoint and available labels. Averaging error over the full acquisition interval also favors iMQS over random pool on each endpoint, although gains remain variable across splits (Section D.1).

Improvements extend to other prediction metrics. iMQS has the lowest mean MSE on all five regression endpoints (Table 2). Classification accuracy is more mixed: iMQS leads on CYP1A2 and exceeds random pool on all four endpoints, while entropy sampling leads on Pgp and SR-MMP, and GOIMDA on CYP3A4. Prediction loss and class decisions therefore give complementary comparisons.

5.3 Molecular screening with docking labels

The benchmark results measure agreement with the labeling oracle. To assess the value of acquired data for screening, we next evaluate recovery of molecules with measured biological activity. We

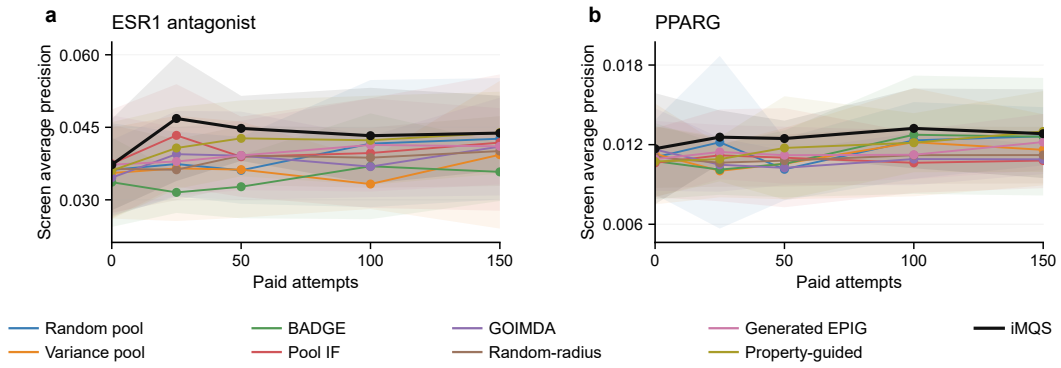


Figure 3: Recovery of experimentally measured screening actives after learning from AutoDock Vina scores. Higher screen average precision is better. Means and sample-SD bands use ten splits.

use LIT-PCBA cell-based functional reporter assays for estrogen receptor alpha antagonism (ESR1 antagonist) and peroxisome proliferator-activated receptor gamma activation (PPARG) (Tran-Nguyen et al., 2020). AutoDock Vina supplies continuous docking scores (Eberhardt et al., 2021); the predictor learns these scores and ranks a disjoint screening library, placing lower predicted scores first. Screen average precision summarizes precision as measured actives are recovered along the whole ranking; higher is better. Measured labels are used for partitioning and evaluation, while acquisition and training use only Vina scores.

The screens contain 55 actives among 2,681 molecules for ESR1 antagonist (2.05%) and 15 among 2,870 for PPARG (0.52%) in every split. We compare nine methods over 0–150 paid acquisition attempts after the same initial and reference labeling setup. Failures consume an attempt. Screening checkpoints are recorded at 0, 25, 50, 100, and 150; straight segments join them. Section C.5 gives receptor preparation, filtering, and ranking details.

Generated training data support screening. iMQS improves mean screen average precision from initialization on both targets (Figure 3). At the final budget, it ranks second to property-guided generation, with a small difference in their means. Random pool is a strong baseline on both targets, with BADGE also competitive on PPARG. Variance sampling, Pool IF, GOIMDA, Random-radius, and Generated EPIG all finish below iMQS in mean performance. This study focuses on acquiring training data to improve model performance; the docking application evaluates the resulting predictor's screening performance. Prospective discovery and experimental confirmation of new biological actives remain future work.

6 Conclusion

We propose iMQS to generate molecular queries for their value as training data under a limited evaluation budget. The method estimates how an additional observation would reduce reference prediction loss and uses this influence score to guide search in a continuous molecular representation. The decoded molecule is evaluated by the oracle, and its observed label updates the predictor, including when it differs from the label requested during search.

Our analysis relates score differences to validation gains after finite, possibly inexact head refits with the earlier layers fixed. The bound accounts for curvature changes, fitting residuals, numerical error, and disagreement between requested and observed labels. Across nine molecular endpoints, iMQS reduces mean test error from initialization. It achieves the lowest mean MSE on all five regression endpoints, while the accuracy leaders differ across classification endpoints. In two docking applications, mean recovery of measured screening actives improves during acquisition and refitting, with benefits that vary across splits and comparison methods. These findings support molecular query synthesis as a way to acquire useful training data. Extending the evaluation to experimental assays and testing whether acquired molecule–label pairs benefit other predictors would help establish where this approach can improve data collection for molecular discovery.

AI USE STATEMENT

OpenAI Codex assisted with manuscript editing, mathematical argument review, coding, and figure preparation. AI-assisted outputs were checked against cited sources, archived experiment records, and executable consistency tests. TxGemma provides the computational labels used in the benchmark experiments, as described in Section 5. The authors are responsible for the final text, claims, code, and analyses.

ETHICS STATEMENT

This study uses public molecular data and computational evaluations, with no new human or animal experiments. Predicted properties and docking scores do not establish clinical safety or efficacy. Generated molecules require appropriate chemical and biological assessment before practical use. The distributed materials retain the applicable data, model, and software license terms.

REPRODUCIBILITY STATEMENT

Sections A and 3 specify the algorithm, and Section B states the assumptions and proofs. Section C describes the empirical protocol. The anonymous source archive includes frozen data roles, configurations, selected run records, and scripts for the learning curves and paired tests; Section C.6 summarizes the available records.

REFERENCES

Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988. doi: 10.1023/A:1022821128753. URL https://doi.org/10.1023/A:1022821128753.

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2020.

Juhan Bae, Nathan Ng, Alston Lo, Marzyeh Ghassemi, and Roger B. Grosse. If influence functions are the answer, then what is the question? In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1305.

Samyadeep Basu, Phillip Pope, and Soheil Feizi. Influence functions in deep learning are fragile. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=xHKVVHGDOEk.

Freddie Bickford Smith, Andreas Kirsch, Sebastian Farquhar, Yarin Gal, Adam Foster, and Tom Rainforth. Prediction-oriented bayesian active learning. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 7331–7348, 2023. URL https://proceedings.mlr.press/v206/bickfordsmith23a.html.

Gavin C. Cawley and Nicola L. C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(70):2079–2107, 2010. URL https://jmlr.org/papers/v11/cawley10a.html.

Jerome Eberhardt, Diogo Santos-Martins, Andreas F. Tillack, and Stefano Forli. AutoDock Vina 1.2.0: New docking methods, expanded force field, and Python bindings. *Journal of Chemical Information and Modeling*, 61(8):3891–3898, 2021. doi: 10.1021/acs.jcim.1c00203.

Peter Eckmann, Kunyang Sun, Bo Zhao, Mudong Feng, Michael K. Gilson, and Rose Yu. LIMO: Latent inceptionism for targeted molecule generation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5777–5792, 2022. URL https://proceedings.mlr.press/v162/eckmann22a.html.

Peter Eckmann, Dongxia Wu, Germano Heinzelmann, Michael K. Gilson, and Rose Yu. MF-LAL: Drug compound generation using multi-fidelity latent space active learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 14972–14988, 2025.

Jillian Fisher, Lang Liu, Krishna Pillutla, Yejin Choi, and Zaid Harchaoui. Influence diagnostics under self-concordance. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 10028–10076, 2023. URL https://proceedings.mlr.press/v206/fisher23a.html.

Ryan Giordano, William Stephenson, Runjing Liu, Michael Jordan, and Tamara Broderick. A swiss army infinitesimal jackknife. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1139–1147, 2019. URL https://proceedings.mlr.press/v89/giordano19a.html.

Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018. doi: 10.1021/acscentsci.7b00572.

David E. Graff, Eugene I. Shakhnovich, and Connor W. Coley. Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chemical Science*, 12:7866–7881, 2021. doi: 10.1039/D0SC06805E. URL https://doi.org/10.1039/D0SC06805E.

Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained bayesian optimization for automatic chemical design using variational autoencoders. *Chemical Science*, 11(2):577–586, 2020. doi: 10.1039/C9SC04026A. URL https://doi.org/10.1039/C9SC04026A.

Yuzheng Hu, Pingbang Hu, Han Zhao, and Jiaqi W. Ma. Most influential subset selection: Challenges, promises, and beyond. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2409.18153.

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W. Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/4c56ff4ce4aaf9573aa5dff913df997a-Abstract-round1.html.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. GLISTER: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8110–8118, 2021. doi: 10.1609/aaai.v35i9.16988.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1885–1894. PMLR, 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

Pang Wei Koh, Kai-Siang Ang, Hubert Teo, and Percy Liang. On the accuracy of influence functions for measuring group effects. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alán Aspuru-Guzik. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020. doi: 10.1088/2632-2153/aba947.

Gregory W. Kyro, Anton Morgunov, Rafael I. Brent, and Victor S. Batista. ChemSpaceAL: An efficient active learning methodology applied to protein-specific molecular generation. *Journal of Chemical Information and Modeling*, 64(3):653–665, 2024. doi: 10.1021/acs.jcim.3c01456.

Zhuoming Liu, Hao Ding, Huaping Zhong, Weijia Li, Jifeng Dai, and Conghui He. Influence selection for active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9274–9283, 2021.

Yasmine Nahal, Janosch Menke, Julien Martinelli, Markus Heinonen, Mikhail Kabeshov, Jon Paul Janet, Eva Nittinger, Ola Engkvist, and Samuel Kaski. Human-in-the-loop active learning for goal-oriented molecule generation. *Journal of Cheminformatics*, 16(1):138, 2024. doi: 10.1186/s13321-024-00924-y.

Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the 18th International Conference on Machine Learning*, pp. 441–448. Morgan Kaufmann, 2001.

Ittai Rubinstein and Samuel B. Hopkins. On the accuracy of newton step and influence function data attributions. In *Proceedings of the International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2512.12572.

Raphael Schumann and Ines Rehbein. Active learning via membership query synthesis for semi-supervised sentence classification. In *Proceedings of the 23rd Conference on Computational Natural Language Learning*, pp. 472–481, 2019. doi: 10.18653/v1/K19-1044.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

Jing Tang, Agnieszka Szwajda, Sushil Shakyawar, Tao Xu, Petteri Hintsanen, Krister Wennerberg, and Tero Aittokallio. Making sense of large-scale kinase inhibitor bioactivity data sets: A comparative and integrative analysis. *Journal of Chemical Information and Modeling*, 54(3):735–743, 2014. doi: 10.1021/ci400709d.

Viet-Khoa Tran-Nguyen, Celien Jacquemard, and Didier Rognan. LIT-PCBA: An unbiased data set for machine learning and virtual screening. *Journal of Chemical Information and Modeling*, 60(9):4263–4273, 2020. doi: 10.1021/acs.jcim.0c00155.

Derek van Tilborg and Francesca Grisoni. Traversing chemical space with active deep learning for low-data drug discovery. *Nature Computational Science*, 4:786–796, 2024. doi: 10.1038/s43588-024-00697-2.

Eric Wang, Samuel Schmidgall, Paul F. Jaeger, Fan Zhang, Rory Pilgrim, Yossi Matias, Joelle Barral, David Fleet, and Shekoofeh Azizi. TxGemma: Efficient and agentic LLMs for therapeutics. *arXiv preprint arXiv:2504.06196*, 2025. URL https://arxiv.org/abs/2504.06196.

Dennis Wei, Inkit Padhi, Soumya Ghosh, Amit Dhurandhar, Karthikeyan Natesan Ramamurthy, and Maria Chang. Final-model-only data attribution with a unifying view of gradient-based methods. In *Advances in Neural Information Processing Systems*, volume 38, pp. 106802–106836, 2025. doi: 10.52202/085713-3562. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/99d7326032bbed26de1b244beaff6a84-Paper-Conference.pdf.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: A benchmark for molecular machine learning. *Chemical Science*, 9:513–530, 2018. doi: 10.1039/C7SC02664A. URL https://pubs.rsc.org/en/content/articlehtml/2018/sc/c7sc02664a.

Weichi Yao, Bianca Dumitrascu, Bryan R. Goldsmith, and Yixin Wang. Goal-oriented influence-maximizing data acquisition for learning and optimization. *arXiv preprint arXiv:2602.19578*, 2026. doi: 10.48550/arXiv.2602.19578.

Jia-Jie Zhu and José Bento. Generative adversarial active learning. *arXiv preprint arXiv:1702.07956*, 2017. doi: 10.48550/arXiv.1702.07956. URL https://arxiv.org/abs/1702.07956.

A Derivation of the acquisition score

This appendix derives the influence score in equation 6 and gives its explicit head derivatives. It applies the standard influence derivative to the addition of one record, accounting for the resulting change in the normalization of the mean training loss.

Lemma A.1. (Standard implicit-differentiation form; cf. Koh & Liang, 2017, Section 2.1 and Appendix A.) *Let F and h be twice continuously differentiable near an interior point $\bar{\theta}$ with $\nabla F(\bar{\theta}) = 0$ and $H = \nabla^2 F(\bar{\theta}) \succ 0$. For sufficiently small ϵ, let $\theta(\epsilon)$ be the local stationary branch of $F + \epsilon h$ through $\theta(0) = \bar{\theta}$. Then this branch is differentiable at zero and*

$$\theta'(0) = -H^{-1} \nabla h(\bar{\theta}). \tag{8}$$

For a continuously differentiable validation loss V, the derivative of $V(\theta(\epsilon))$ at zero is $-\nabla V(\bar{\theta})^\top H^{-1} \nabla h(\bar{\theta})$.

Proof. The implicit function theorem gives the local stationary branch because H is invertible. Differentiating $\nabla F(\theta(\epsilon)) + \epsilon \nabla h(\theta(\epsilon)) = 0$ at zero yields $H\theta'(0) + \nabla h(\bar{\theta}) = 0$. The validation derivative follows by the chain rule. □

This is the usual influence calculation with a general additive perturbation. We apply it with h equal to the candidate loss minus the current mean training loss, accounting for the normalization change when one record is added.

A.1 Adding a record to a mean-loss objective

Hold the predictor body fixed and write $F_t(\theta) = L_t(\omega_t, \theta) + \lambda \|\theta\|^2 / 2$. To describe the addition of (z, y^*), increase its weight continuously while reducing the weight of the old mean loss:

$$\theta_{t,\epsilon}(z, y^*) = \arg\min_{\theta} \left\{ F_t(\theta) + \epsilon \big[\ell(\omega_t, \theta; z, y^*) - L_t(\omega_t, \theta) \big] \right\}, \qquad \epsilon \in [0, 1]. \tag{9}$$

At $\epsilon = (n_t + 1)^{-1}$, the objective is exactly the mean loss on the augmented set plus the unchanged regularizer. For a fixed linear head, cross-entropy and squared-error loss are convex in θ; the positive regularizer makes their objectives strongly convex. The minimizer is therefore unique.

For this derivative calculation, suppose θ_t minimizes F_t. Applying Lemma A.1 with $h = \ell(\omega_t, \cdot; z, y^*) - L_t(\omega_t, \cdot)$ gives

$$\left. \frac{\mathrm{d}\theta_{t,\epsilon}(z, y^*)}{\mathrm{d}\epsilon} \right|_{\epsilon=0} = -\widehat{H}_t^{-1} \big[g_t(z, y^*) - \nabla_\theta L_t(\omega_t, \theta_t) \big]. \tag{10}$$

The validation-loss derivative is consequently

$$\left. \frac{\mathrm{d}V_t(\theta_{t,\epsilon}(z, y^*))}{\mathrm{d}\epsilon} \right|_{\epsilon=0} = -v_t^\top \widehat{H}_t^{-1} \big[g_t(z, y^*) - \nabla_\theta L_t(\omega_t, \theta_t) \big]. \tag{11}$$

A first-order expansion at zero and the substitution $\epsilon = (n_t + 1)^{-1}$ predict validation-loss reduction $(A_t(z, y^*) - \kappa_t)/(n_t + 1)$ when the linear solve is exact, with A_t defined in equation 6. Here $\kappa_t = \widehat{u}_t^\top \nabla_\theta L_t(\omega_t, \theta_t)$. Rescaling the old records produces this candidate-independent offset, which can be omitted when ranking queries. The derivative is evaluated at a stationary head; Theorem B.2 separately treats finite, inexact refits and numerical solve error at the implemented checkpoint.

A.2 Explicit head derivatives

For the classifier, let $a_t(z) = [\phi_{\omega_t}(z)^\top, 1]^\top$ include the bias coordinate, let $p_t(z) = f_{\omega_t, \theta_t}(z)$, and let e_y denote the standard basis vector for class y. The gradient is

$$g_t(z, y) = \nabla_\theta \ell(\omega_t, \theta_t; z, y) = \mathrm{vec}_{\mathrm{row}} \big[(p_t(z) - e_y) a_t(z)^\top \big], \tag{12}$$

where $\mathrm{vec}_{\mathrm{row}}$ stacks matrix rows. With identity matrix I, diagonal operator Diag, and Kronecker product $\otimes$, the regularized Hessian is

$$W_t(z) = \mathrm{Diag}(p_t(z)) - p_t(z) p_t(z)^\top, \tag{13}$$

$$\widehat{H}_t = \frac{1}{n_t} \sum_i W_t(z_i) \otimes a_t(z_i) a_t(z_i)^\top + \lambda I. \tag{14}$$

These are the usual softmax cross-entropy derivatives. For a scalar head, use the half squared-error loss $\ell(\omega, \theta; z, y) = \frac{1}{2}(h_\theta(\phi_\omega(z)) - y)^2$. The corresponding expressions are

$$g_t(z, y) = (h_{\theta_t}(\phi_{\omega_t}(z)) - y) a_t(z), \qquad \widehat{H}_t = \frac{1}{n_t} \sum_i a_t(z_i) a_t(z_i)^\top + \lambda I. \tag{15}$$

In either case, training data determine the Hessian and reference data determine v_t. Solving $\widehat{H}_t \widehat{u}_t = v_t$ gives the vector used for every candidate in the cycle.

A.3 Search derivatives and fitting conventions

Differentiating this score with respect to the candidate feature gives the search direction:

$$\nabla_z A_t(z, y^*) = \left[\frac{\partial g_t(z, y^*)}{\partial z} \right]^\top \widehat{u}_t. \tag{16}$$

The Jacobian differentiates the candidate gradient through its predicted probabilities and hidden features. During search, the predictor, labeled and reference data, and $\widehat{u}_t$ remain fixed. Thus latent optimization requires neither repeated refits nor derivatives through the discrete decoder.

For classification, ties between least represented classes follow a cyclic order, starting at class zero and advancing past each chosen class. For regression, each standardized training response is assigned to its nearest reference value in $\{-1, 0, 1\}$; the least populated group determines the requested response. The same cyclic rule breaks ties. The chosen label remains fixed for the cycle. The full MLP is refitted at each acquisition cycle using Adam with coupled L2 coefficient $\lambda > 0$ on every weight and bias, corresponding to $L_t + \lambda \|(\omega, \theta)\|^2 / 2$.

A.4 Reference loss and test performance

For a predictor fixed independently of a validation sample, the sample's average loss is an unbiased estimate of expected loss on new observations from the same joint distribution of molecules and labels. This motivates using reference loss to value training data. The acquisition procedure repeatedly uses the reference set to choose new training examples, so the resulting predictor depends on that set. Its reference loss is therefore a selection objective, rather than an unbiased report of test performance after adaptive selection (Cawley & Talbot, 2010). The separate test set supplies the performance evaluation. In the docking application, measured screening activities evaluate the ranking produced by a predictor trained on docking scores.

B Proofs and supporting analysis

This appendix gives the assumptions and explicit error term for Theorem 4.1. The proof first compares the fitted heads, then transfers that comparison to validation loss and candidate selection.

B.1 Finite query-ranking bound

Fix one acquisition cycle, hold the predictor body fixed, and suppress the cycle index: $n = n_t \geq 1$, $\theta_0 = \theta_t$, $V = V_t$, $H = \widehat{H}_t = \nabla^2 F_0(\theta_0)$, $v = \nabla V(\theta_0)$, and $s = \widehat{u}_t$. The vector s is the returned numerical solution, so $Hs = v$ need not hold exactly. Let $\mathcal{C}$ be a nonempty finite set of eligible candidates $q = (z_q, y_q)$, where y_q is the oracle label in the coordinates used for training. Write $S(z) = A_t(z, y^*) = g_t(z, y^*)^\top s$ for the computed score under the cycle's fixed requested label. With $L_n(\theta) = L_t(\omega_t, \theta)$ and $a = (n+1)^{-1}$, the addition objective from equation 9 is

$$F_0 = L_n + \frac{\lambda}{2}\|\theta\|^2, \qquad F_q = F_0 + \psi_q, \qquad \psi_q = a[\ell(\theta; z_q, y_q) - L_n(\theta)], \quad \lambda > 0. \tag{17}$$

Fitting F_q returns a head θ_q with validation gain $\Delta_q = V(\theta_0) - V(\theta_q)$ and residual $\rho_q = \nabla F_q(\theta_q)$, which vanishes at an interior optimum. Set $g_q = \nabla \psi_q(\theta_0)$ and $g_q^* = a[\nabla \ell(\theta_0; z_q, y^*) - \nabla L_n(\theta_0)]$. These are the scaled perturbation gradients for the observed and requested labels. The computed score predicts the gain

$$\widehat{T}_q = s^\top g_q^* = \frac{A_t(z_q, y^*) - \kappa_t}{n_t + 1} = aS(z_q) - a\kappa_t, \qquad \kappa_t = s^\top \nabla L_n(\theta_0). \tag{18}$$

Thus the positive scale a and common offset κ_t preserve score rankings. However, the oracle label may differ from the requested label. Its score correction b_q enters the comparison as

$$b_q = s^\top[\nabla \ell(\theta_0; z_q, y_q) - \nabla \ell(\theta_0; z_q, y^*)], \qquad s^\top(g_q - g_r) = a[S(z_q) - S(z_r) + b_q - b_r]. \tag{19}$$

This accounts for label changes in the linear prediction. To control its remaining error after a finite refit, we measure displacement and curvature relative to H (Fisher et al., 2023): $\|u\|_H = (u^\top H u)^{1/2}$ for parameter vectors, $\|w\|_* = (w^\top H^{-1} w)^{1/2}$ for gradients, and $\|B\|_{\mathrm{rel}} = \left\|H^{-1/2} B H^{-1/2}\right\|_2$ for symmetric matrices, where $\|\cdot\|_2$ is the operator norm.

Assumption B.1. *The matrix H is positive definite. The functions F_0, F_q, V are twice continuously differentiable on a neighborhood of a convex region containing θ_0 and every θ_q. There are finite constants $D, L_H, \beta_q, k_{qr} \geq 0$ and $\gamma > 0$ such that $\|\theta_q - \theta_0\|_H \leq D$ for all $q \in \mathcal{C}$ and, uniformly on this region for all $q, r \in \mathcal{C}$,*

$$\begin{aligned} &\nabla^2 F_q \succeq \gamma H, && \left\|\nabla^2 F_q - H\right\|_{\mathrm{rel}} \leq \beta_q, && \gamma > 0, \\ &\left\|\nabla^2 V\right\|_{\mathrm{rel}} \leq L_H, && \left\|\nabla^2(\psi_q - \psi_r)\right\|_{\mathrm{rel}} \leq k_{qr}, && k_{qq} = 0. \end{aligned} \tag{20}$$

The lower curvature bound makes every mean Hessian in the proof invertible. The other bounds control the error incurred when replacing finite changes by derivatives at θ_0. No stationarity assumption is imposed on θ_0 or the returned refits.

Put $C_V = \|v\|_*$ and denote the numerical solve residual by $e = Hs - v$. Define

$$\begin{aligned} d_{qr} &= g_q - g_r, & p_{qr} &= \|d_{qr}\|_*, \\ \zeta_{qr} &= \|\rho_q - \rho_r\|_*, & \sigma_{qr} &= |e^\top H^{-1} d_{qr}|. \end{aligned} \tag{21}$$

Here p_{qr} and ζ_{qr} measure differences in query gradients and fitting residuals; σ_{qr} is the solve error along the pairwise influence response. Set $\beta_{qr} = \min\{\beta_q, \beta_r\}$ and

$$\mathcal{E}_{qr} = \frac{(C_V\beta_{qr} + L_H D)p_{qr} + (C_V + L_H D)(k_{qr}D + \zeta_{qr})}{\gamma} + \sigma_{qr}. \tag{22}$$

The combined bound used in the main paper is

$$\mathcal{B}_{qr} = \mathcal{E}_{qr} + a|b_q - b_r|. \tag{23}$$

It includes both the finite-refit error and the error in replacing the observed label by the requested label. In particular, $\mathcal{B}_{qq} = 0$.

Theorem B.2. *Under Assumption B.1, every pair $q, r \in \mathcal{C}$ satisfies*

$$|(\Delta_q - \Delta_r) - a[S(z_q) - S(z_r)]| \le \mathcal{E}_{qr} + a|b_q - b_r|. \tag{24}$$

For any returned candidate $\widehat{q} \in \mathcal{C}$, the gain lost relative to the best candidate obeys

$$\max_{q\in\mathcal{C}} \Delta_q - \Delta_{\widehat{q}} \le \max_{q\in\mathcal{C}} \{a[S(z_q) - S(z_{\widehat{q}}) + b_q - b_{\widehat{q}}] + \mathcal{E}_{q\widehat{q}}\}. \tag{25}$$

Here the $q = \widehat{q}$ term is zero. The bounds are unchanged by an invertible affine reparameterization of the same objectives, fitted heads, and numerical solution.

B.2 Proof of Theorem B.2

We adapt the mean-Hessian and inverse-matrix comparison of Fisher et al. (2023, Appendix E.2). We compare two refits directly, retaining their fitting residuals, and transfer the parameter comparison to validation loss. The argument uses the stated local assumptions rather than statistical concentration bounds.

Proof of Theorem B.2. Fix q, r and write $d = d_{qr}$, $\delta = \theta_q - \theta_r$, and $u_r = \theta_r - \theta_0$. Define the mean Hessian between refits and the remainder in their gradient difference:

$$B_q = \int_0^1 \nabla^2 F_q(\theta_r + t\delta)\, dt, \qquad t_{qr} = \int_0^1 \nabla^2(\psi_q - \psi_r)(\theta_0 + tu_r)u_r\, dt.$$

Since $F_q - F_r = \psi_q - \psi_r$, the fundamental theorem of calculus gives

$$\begin{aligned} B_q\delta &= \nabla F_q(\theta_q) - \nabla F_q(\theta_r) \\ &= \rho_q - \rho_r - \nabla(\psi_q - \psi_r)(\theta_r) = \rho_q - \rho_r - d - t_{qr}. \end{aligned} \tag{26}$$

The common gradient at θ_0 is absent from this difference equation; the effect of the shared objective's curvature remains in B_q. The theorem's bound on the query-curvature difference gives $\|t_{qr}\|_* \le k_{qr}D$.

To use the norms in the theorem, express the average curvature on the scale set by H: put $\widetilde{B}_q = H^{-1/2}B_qH^{-1/2}$. Averaging the curvature assumptions gives $\widetilde{B}_q \succeq \gamma I$ and $\left\|\widetilde{B}_q - I\right\|_2 \le \beta_q$. Applying these bounds to equation 26 first controls the distance between the two refits:

$$\|\delta\|_H \le \frac{p_{qr} + k_{qr}D + \zeta_{qr}}{\gamma}. \tag{27}$$

Influence predicts the parameter difference $-H^{-1}d$. To compare it with δ, rearrange equation 26 into the exact inverse identity

$$\delta + H^{-1}d = B_q^{-1}[(B_q - H)H^{-1}d - t_{qr} + \rho_q - \rho_r]. \tag{28}$$

The terms on its right describe curvature change, the query-difference remainder and unequal fitting residuals. Multiplying by $H^{1/2}$, using $\left\|\widetilde{B}_q^{-1}\right\|_2 \le 1/\gamma$, and observing that $\left\|H^{-1}d\right\|_H = p_{qr}$ yields

$$\left\|\delta + H^{-1}d\right\|_H \le \frac{\beta_q p_{qr} + k_{qr}D + \zeta_{qr}}{\gamma}. \tag{29}$$

Incomplete fitting at the original checkpoint still affects the displacement bound D.

To translate this parameter comparison into validation gain, note that the segment between refits lies within distance D of θ_0. Hence validation curvature bounds the gradient change by $L_H D$ in the dual norm. Integrating gives

$$|(\Delta_q - \Delta_r) + v^\top \delta| = \left| \int_0^1 [\nabla V(\theta_r + t\delta) - v]^\top \delta \, dt \right| \leq L_H D \, \|\delta\|_H .$$

Combining this with equation 29 bounds the error relative to $v^\top H^{-1} d$. To reach the implemented score, we account for the numerical solve through the exact identity

$$s^\top d - v^\top H^{-1} d = (Hs - v)^\top H^{-1} d.$$

Taking its absolute value gives σ_{qr} rather than a bound involving the entire solve-residual norm. Interchanging q, r proves the same estimate with β_r, with the same k_{qr}, since reversing the Hessian difference leaves its norm unchanged. The smaller bound is therefore valid. This establishes $|(\Delta_q - \Delta_r) - s^\top d_{qr}| \leq \mathcal{E}_{qr}$. Substituting equation 19 proves equation 24. Using the signed upper bound for every q against $\widehat{q}$ and then maximizing proves equation 25.

For Theorem 4.1, replacing the signed label correction in equation 25 by its absolute value and using equation 23 gives equation 7. If $\widehat{q}$ maximizes the score over $\mathcal{C}$, every score difference in that maximum is nonpositive, proving the final claim.

Finally, to establish coordinate invariance, we use the transformation rule for Hessian norms in Fisher et al. (2023, Section 3). Write $\theta = P\xi + c$, where ξ denotes the new coordinates, P is invertible and c is fixed. Transforming the same objectives gives $H_\xi = P^\top H P$, $g_\xi = P^\top g$, $u_\xi = P^{-1} u$ and $s_\xi = P^{-1} s$. Hence all primal and dual norms and the scalar $e^\top H^{-1} d$ are unchanged. For symmetric Hessians, the relative norm equals the largest absolute generalized eigenvalue of the pair (B, H), which is also unchanged under this congruence. The curvature inequalities, radius and error bound therefore agree in both coordinates. The regularizer is transformed as part of the same objective; replacing it by a new isotropic penalty would change the problem. □

Relation to prior work. The influence derivative follows Koh & Liang (2017). The proof adapts the deterministic mean-Hessian and inverse-matrix comparison in Fisher et al. (2023, Appendix E.2), whose Theorem 2 controls a single finite perturbation around an optimized estimator. Shared effects of incomplete fitting appear in Koh & Liang (2017, Appendix B), and Wei et al. (2025, Section 3) adjust attribution for further training on unchanged data. Our extension compares two finite, possibly inexact refits from the same checkpoint, retaining candidate differences to bound ranking error and selection regret.

C Experimental details

This appendix specifies the data roles, comparator adaptations, and definitions needed to interpret the reported comparisons. The accompanying source archive contains the selected dataset configurations, inference rules, search and fitting parameters, and records used to reproduce the figures and tables.

C.1 Molecular endpoints and teacher labels

The binary tasks cover inhibition of cytochrome P450 enzymes 1A2 and 3A4 (CYP1A2 Veith and CYP3A4 Veith), P-glycoprotein inhibition (Pgp Broccatelli), and mitochondrial membrane potential disruption in the Toxicology in the 21st Century program (Tox21 SR-MMP). The regression tasks comprise kinase inhibitor bioactivity (KIBA) scores for glycogen synthase kinase 3 beta (GSK3B), protein kinase A catalytic subunit alpha (PRKACA), and vascular endothelial growth factor receptor 2 (VEGFR2); acute toxicity expressed through median lethal dose (LD50 Zhu); and lipophilicity (Lipophilicity AstraZeneca). The underlying datasets supply molecular structures. The teacher supplies the labels used to train and evaluate the benchmark predictors.

Eligibility requires valid, nonconflicting source labels, canonical molecular identities, and compatibility with the frozen VAE. Preparation excludes a fixed 20% teacher-diagnostic reserve before assigning learner roles. Within the remaining cohort, each random split allocates 20% to testing, two

ordered reservoirs of up to 200 molecules each to initialization and reference evaluation, and the rest to the candidate pool. Membership is not stratified by endpoint labels. The first 100 records with valid teacher responses in each reservoir supply the initial and reference sets. Regression role maps exclude unavailable teacher responses from the labeled reservoirs and query pool, while preserving a structural pool for molecular identity checks. Frozen memberships and preparation rules are supplied with the source. Held-out identities are excluded from acquisition.

TxGemma-9B-Predict receives the endpoint-specific prompt and canonical molecular string under a fixed inference rule. For binary tasks, the normalized probabilities of the two allowed answers define the teacher score; the more probable class supplies the training and test label. Regression prompts request an integer from 0 to 1000, which is divided by 1000 to obtain a unit-interval teacher score. Initial-label mean and standard deviation separately standardize the learner's targets, and predictions are transformed back to the teacher-score scale before computing test MAE. This score has no physical inverse transform, so MAE is not expressed in assay units. Test NLL and MAE are averaged over the fixed test records with valid teacher responses and measure agreement with the labeling model. Agreement between the teacher and measured labels is evaluated separately in Section D.2.

The final-budget comparison in Table 2 uses the same fixed test records and teacher labels as the learning curves. Accuracy is the fraction of teacher labels matched by the student's highest-probability class. MSE averages squared differences after predictions are transformed back to the unit-interval teacher-score scale; table entries are multiplied by 10^3. Each cell reports the mean and sample standard deviation of ten split-level metrics, with equal weight per split. The source file `data/reported/additional_test_metrics.csv` and `scripts/summarize_additional_metrics.py` reproduce the table summaries. The table uses the same comparator configurations as Table 1; the additional lower-response comparisons are included in the source file `results/additional_metrics/main_teacher_endpoint_summary_lower_response.csv`.

C.2 Predictors and comparator adaptations

The shared evaluation predictor uses the frozen LIMO encoder and an MLP with one 64-unit hidden layer, no dropout, and a linear classification or regression head. It is reset and trained on all available labeled records at each acquisition cycle, with mean cross-entropy or half squared-error loss and coupled L2 coefficient $\lambda = 0.01$. Reference loss determines early stopping and supplies the acquisition gradient. The source file `src/almqs/config/studies.py` defines the fitting and search parameters in `FIT` and `SEARCH`. In influence search, derivatives are restricted to the final head; the full MLP is subsequently retrained. Optimized raw features are retained for iMQS, property-guided search, and the matched random-direction control. Pool molecules and native-generator queries use encoder features. Generated EPIG scores and stores encoder features on eight teacher endpoints, and proposal features on CYP3A4. In docking it scores encoder features but stores proposal features. Thus Random-radius and property-guided search use the same training-feature convention as iMQS throughout.

Pool IF chooses the highest influence score among eligible library molecules with the same requested-label rule as iMQS. Entropy sampling uses predictive entropy in classification and ensemble variance in regression. BADGE uses gradient embeddings for classification; its single-query regression adaptation weights ensemble variance by the squared predictor-body feature norm. GOIMDA uses expected head influence, with class probabilities or a mean regression response supplied by a five-model bootstrap ensemble. This ensemble is separate from the evaluation predictor whose head gradient is scored. Generated EPIG scores candidate molecules by their information about predictions on the fixed reference inputs. Its candidate construction uses random latent directions with distances matched to influence search. Random-radius draws such a direction without EPIG selection. Property-guided search maximizes the predictor's preferred-class probability or the configured continuous-response utility using the same latent search machinery.

Nahal-EPIG retains the REINVENT generator and random-forest acquisition predictor of Nahal et al. (2024); the shared MLP evaluates its acquired labels. Its generator reward and sampling procedure differ from those of the VAE methods. On CYP3A4, its generator reward uses the positive-class probability, whereas property-guided latent search favors noninhibition. These comparator objectives

determine the queried molecules; prediction error evaluates the resulting training data. On KIBA and Lipophilicity, both methods use separate higher- and lower-response preferences. These exploratory directions refer to the normalized teacher response, not a biological optimum. For docking, all nine methods use their continuous-response adaptations.

For continuous responses, we approximate EPIG by averaging the Gaussian mutual information $-\frac{1}{2}\log(1-c^2)$ over reference inputs. Here c is the candidate–reference prediction correlation across ensemble members, with a common noise variance added to both marginal variances. Generated EPIG uses five bootstrap MLPs, setting the noise variance to the mean squared error of their average reference prediction on standardized responses. Nahal-EPIG uses random-forest tree predictions and the initial response variance on the teacher-score scale. Both noise variances have a floor of 10^{-6}.

C.3 Acquisition accounting and curve aggregation

Each split shares 100 initial and 100 reference labels across methods; zero on the acquisition axis follows this setup. Binary benchmarks and docking count all oracle requests, including failures and cached responses; regression counts successful labels. Proposals rejected before evaluation consume no budget. Initial, reference, held-out, and previously queried identities are excluded from acquisition. Failed calls supply no label. Regression retries failures within the same cycle; docking advances the cycle and resets and refits the predictor even when a failure leaves its training data unchanged.

Every benchmark method has ten runs per setting (splits 0–9). Binary acquisition calls all succeed, so attempts and labels coincide. Regression records contain 180 failed calls, with at most 155 calls per 150-label trajectory; equal label budgets therefore need not have equal paid cost. The 980 benchmark trajectories cover ten methods on nine endpoints, including separate higher/lower settings for two methods on four endpoints. The other eight methods share trajectories across those directions. Docking search limits are fixed across runs.

Curves show means and sample standard deviations across splits at each recorded acquisition count. Benchmarks record every count; docking records 0, 25, 50, 100, and 150 paid attempts, joined by straight segments without smoothing or extrapolation. Shared source molecules make this repeated-split variability, rather than independent biological replication. Paired tests assess differences at the specified budgets.

C.4 Paired comparisons at fixed acquisition budgets

At budgets 50, 100, and 150, two-sided paired t-tests at $\alpha = 0.05$ compare iMQS with each baseline. Each test pairs ten splits within a dataset and tests whether the mean difference, iMQS minus baseline, is zero. Responses are teacher test NLL for classification and normalized teacher test MAE for regression. Budgets count binary paid attempts or regression labels. Higher- and lower-response comparators are tested separately against the same iMQS trajectory.

A win means significantly lower error for iMQS, a loss means significantly higher error, and a nonsignificant difference means that the test does not reject equal means. Nonsignificance does not establish equivalence. Tests are exploratory and unadjusted across datasets, baselines, and checkpoints; they describe pointwise comparisons, not simultaneous error control. Random splits reuse source molecules, so the results quantify variability across the recorded splits rather than independent biological replication. For Nahal-EPIG and property-guided search, the table uses the higher-response setting on KIBA and Lipophilicity and the fixed preference on the other five endpoints. Each W/NS/L total covers all nine datasets.

The source archive includes every paired test, including lower-response comparisons, in `results/paired_tests/`. The accompanying `paired_effects.csv` reports mean within-split differences and pointwise 95% Student-t intervals, computed as $\bar{d} \pm t_{0.975,9} s_d/\sqrt{10}$. Here $\bar{d}$ and s_d are the mean and sample SD of the ten paired differences, and $t_{0.975,9}$ is the Student-t quantile with nine degrees of freedom. These unadjusted intervals describe split-level differences conditional on the supplied datasets. Classification effects are in NLL units; regression effects are on the unit-interval teacher-score scale. Their magnitudes should be interpreted within each endpoint.

C.5 Docking and measured screening activity

We use LIT-PCBA structures that pass canonicalization and VAE compatibility checks. The ESR1 antagonist endpoint derives from a cell-based estrogen-response luciferase assay (PubChem BioAssay 743080); PPARG uses a cell-based beta-lactamase reporter assay of receptor activation (BioAssay 743094). These functional activity labels evaluate screening, not the docking pose or activity of newly generated molecules. The eligible cohorts contain 3,881 and 4,070 structures, respectively. Each split assigns 100 initial, 100 reference, and 1,000 pool molecules, with the remainder held out for screening. Measured labels stratify these roles. Canonical identities are disjoint across roles and matched across methods. Only the evaluator reads activity labels after fitting; acquisition and training receive Vina scores. All 100 initial and 100 reference calls return valid scores in every reported target–split combination.

AutoDock Vina uses the frozen ESR1 antagonist-bound receptor template 5FQV and PPARG ligand-binding template 3B1M. The binding region surrounds the crystallographic ligand, which is removed before docking. The source package supplies ligand preparation, receptors, box definitions, and executable settings. All queries pass molecular validity and representation checks. Generated proposals additionally satisfy limits on elements, molecular size, rotatable bonds, and ring size; these limits do not apply to pool preparation. Docking search first draws ordinary proposals, then uses a bounded fallback that cycles through re-encoded paid molecules and shortens optimized displacements before decoding. Exhausting both stages ends search; their limits are given in the source. Fallback use was not separately evaluated. Failed preparation or docking calls consume a paid attempt. Lower Vina scores express a computational binding preference, not measured cellular activity.

At 150 paid attempts, runs return between 136 and 150 acquisition labels. Across all 180 runs, 26,470 of 27,000 attempts succeed. The mean acquired-label counts for iMQS are 144.6 on ESR1 and 145.4 on PPARG, compared with 149.9 on each target for random-pool selection and 143.9 and 142.1 for property-guided search. These counts exclude the initial and reference labels. The source file `results/docking_label_counts.csv` gives successful-label and failed-call means and ranges for every target and method.

At each checkpoint, the predictor ranks the complete held-out screen by increasing predicted Vina score. Screen average precision is $\sum_k (R_k - R_{k-1})P_k$, where P_k and R_k are precision and recall after including the kth distinct score group, starting from $R_0 = 0$. Equal predicted scores enter together. We use scikit-learn's `average_precision_score` on negative predicted Vina scores, with activity labeled 1; the archived evaluator and the supplied execution code use this same function. Method-specific deterministic fitting seeds allow different initial predictions despite shared initial molecules. The comparison contains 180 runs: two targets, nine methods, and ten splits. Across these splits, iMQS increases screen average precision from initialization in nine ESR1 and six PPARG runs. Its paired mean changes are 0.0065 ± 0.0069 and 0.0011 ± 0.0055, respectively (sample standard deviation). These descriptive changes combine acquisition with cycle-specific refitting; the study has no unchanged-data refitting control.

C.6 Reproducing the reported results

The anonymous source archive supplies the implementation, configurations, frozen splits, reported observations, query ledgers, and analysis scripts. Its README gives installation and model-asset guidance. Docking proposal vectors are supplied.

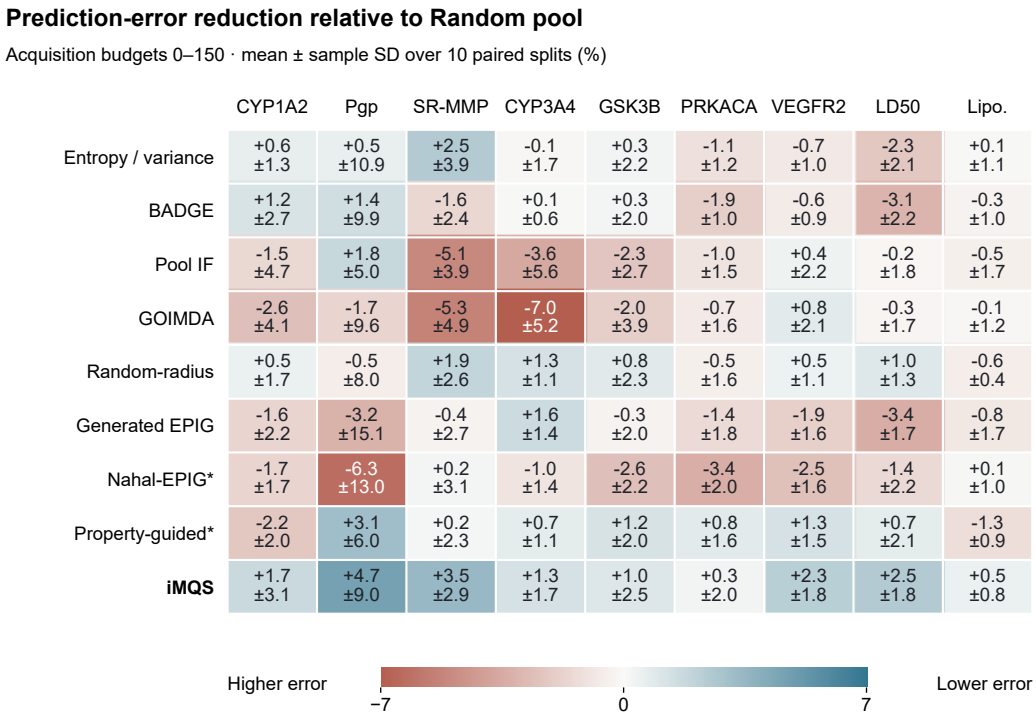


Figure 4: Relative reduction in integrated prediction error versus random pool over budgets 0–150 (%; mean and sample SD over ten paired splits). Positive values favor the listed method. Budgets count binary paid attempts or regression labels. *Higher-response configuration on KIBA and Lipophilicity.

D Additional Experimental Results

D.1 Prediction Error Across the Acquisition Budget

The fixed-budget comparisons show differences at selected checkpoints. We complement them with a retrospective summary that integrates each recorded learning curve over budgets 0–150. For one dataset and split, let $e_m(b)$ be the test error of method m at budget b. Its average error over the interval is

$$\bar{e}_m = \frac{1}{150} \sum_{b=0}^{149} \frac{e_m(b) + e_m(b+1)}{2}. \tag{30}$$

This trapezoidal sum uses every observed benchmark checkpoint. We then calculate $100(\bar{e}_{\text{random}} - \bar{e}_m)/\bar{e}_{\text{random}}$ against random-pool acquisition on the same split. Positive values indicate lower integrated error. Normalizing within each dataset and split keeps classification NLL and regression MAE on their own scales; we do not pool the raw errors. Figure 4 shows the mean and sample standard deviation of these relative reductions over ten splits, using the comparator configurations in Table 1.

The mean relative reduction for iMQS is positive on all nine endpoints, with the largest reductions on Pgp and SR-MMP. It has the largest mean reduction among the displayed methods on six endpoints. The exceptions are CYP3A4, where Generated EPIG and Random-radius have lower mean integrated error, and GSK3B and PRKACA, where Property-guided has lower mean integrated error. Variation across splits remains substantial: on PRKACA, only four of ten splits favor iMQS over random pool despite a positive mean reduction. Thus, the advantage across the acquisition interval is endpoint-dependent and does not hold on every split. These summaries are descriptive; the pointwise tests in Table 1 address a different comparison.

Table 3: TxGemma-9B-Predict agreement with measured endpoint labels on the teacher-diagnostic reserve. Higher agreement metrics are better.

(a) Classification Dataset	Valid / queried	AUROC	Balanced accuracy
CYP1A2 Veith	128/128	0.987	0.945
Pgp Broccatelli	128/128	0.992	0.953
Tox21 SR-MMP	128/128	0.958	0.889
CYP3A4 Veith	2290/2290	0.975	0.922
(b) Regression **Dataset**	**Valid / queried**		**Spearman ρ**
KIBA GSK3B	102/102		0.816
KIBA PRKACA	109/109		0.822
KIBA VEGFR2	128/128		0.845
LD50 Zhu	127/128		0.731
Lipophilicity AstraZeneca	126/128		0.793

D.2 Teacher agreement with measured labels

The learning curves assess how well the acquired data train a predictor to match TxGemma. To assess the teacher itself, we compare its outputs with the original measured endpoint labels on the separate 20% reserve described in Section C.1. These molecules are excluded from the learner's initial, reference, pool, and test sets. The recorded evaluation covers the full reserve for CYP3A4 and fixed diagnostic subsets for the other eight endpoints. Each row of Table 3 describes one fixed audit set, rather than an average over the active-learning splits.

For classification, we report the area under the receiver operating characteristic curve (AUROC) and balanced accuracy, the mean recall of the two classes. For regression, Spearman's rank correlation compares the ordering of the teacher scores with the measured values. This comparison does not require converting the normalized teacher scores into assay units. Metrics use valid responses; one LD50 query and two Lipophilicity queries returned no valid numerical label.

The teacher discriminates the binary labels well on these reserved molecules, although balanced accuracy is lower on SR-MMP than on the other classification endpoints. Its continuous scores are positively correlated with measured responses on all five regression endpoints, with the strongest rank agreement on VEGFR2 and the weakest on LD50. The eight subset evaluations also checked response repeatability by freshly querying the first three diagnostic molecules again, bypassing the cache; all repeated responses matched exactly. This limited repeatability check is separate from agreement with measured labels.

These diagnostics also informed endpoint eligibility, so they characterize the reported benchmarks. The reserve is held out from the active learner, but independence from TxGemma's pretraining data is not established. These results therefore describe teacher agreement on the reserved benchmark molecules, without establishing accuracy on newly generated structures. The saved audit summaries and the script that formats Table 3 are supplied in `data/reported/teacher_diagnostics.json` and `scripts/render_teacher_diagnostics.py`.