Post Hoc Regression Refinement via Pairwise Rankings

Kevin Tirta Wijaya
MPI-INF
kevintirta.w@gmail.com

Michael Sun
MIT
msun415@csail.mit.edu

Minghao Guo
MIT
guomh2014@gmail.com

Hans-Peter Seidel
MPI-INF
hpseidel@mpi-inf.mpg.de

Wojciech Matusik
MIT
wojciech@csail.mit.edu

Vahid Babaei
MPI-INF
vbabaei@mpi-inf.mpg.de

Abstract

Accurate prediction of continuous properties is essential to many scientific and engineering tasks. Although deep-learning regressors excel with abundant labels, their accuracy deteriorates in data-scarce regimes. We introduce `RankRefine`, a model-agnostic, plug-and-play post hoc method that refines regression with expert knowledge coming from pairwise rankings. Given a query item and a small reference set with known properties, `RankRefine` combines the base regressor's output with a rank-based estimate via inverse-variance weighting, requiring no retraining. In molecular property prediction task, `RankRefine` achieves up to 10% relative reduction in mean absolute error using only 20 pairwise comparisons obtained through a general-purpose large language model (LLM) with no finetuning. As rankings provided by human experts or general-purpose LLMs are sufficient for improving regression across diverse domains, `RankRefine` offers practicality and broad applicability, especially in low-data settings.

1 Introduction

Accurate prediction of continuous properties is crucial across scientific and engineering disciplines. In molecular-property prediction (MPP), for instance, reliable estimates of physical or chemical attributes can accelerate drug discovery, materials design, and catalyst development. Recent advances in deep learning have enhanced regression models by leveraging large datasets to uncover complex, nonlinear relationships between structured inputs (e.g., molecular graphs, crystal structures) and their associated properties. Yet, unlike computer vision or natural-language processing, where large labeled corpora can be mined or crowd-sourced, many specialized fields face a fundamental bottleneck: acquiring ground-truth labels requires expert-led experiments that are both costly and slow. Consequently, real-world tasks often operate in data-scarce regimes where even 50 labeled samples may be considered plentiful [Sun et al., 2024].

When large-scale labels are unavailable, expert knowledge becomes a valuable but under-utilized asset. One accessible form of such knowledge is relative comparison: pairwise judgments about which of two samples exhibits a higher (or lower) property value. Pairwise rankings convey substantial information, are often easier for human experts to provide, and can even be generated by general-purpose large language models (LLMs), which have shown surprising ranking skills in chemistry and related domains [Sun et al., 2025, Guo et al., 2023].

We introduce `RankRefine`, a post hoc refinement framework that improves regression predictions by incorporating pairwise rankings. Given a query sample and a small set of labeled references, which can conveniently be drawn from training data, a ranker infers the relative ordering between the query and each reference; these comparisons define a likelihood whose minimization yields a rank-based

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

property estimate. `RankRefine` then fuses this estimate with the base regressor's prediction via inverse-variance weighting, all without retraining.

Because `RankRefine` is largely model-agnostic and requires no architectural changes, it suits low-data, few-shot, and meta-learning scenarios. Leveraging ranking signals from publicly hosted LLMs, it provides substantial improvements in regression accuracy with minimal labeled data and negligible client-side computation. Experiments on synthetic and real-world benchmarks, including multiple MPP tasks, demonstrate that `RankRefine` consistently boosts predictive performance. Further analysis links its effectiveness to ranking quality, underscoring the promise of LLMs and other expert-informed rankers as practical complements to data-scarce regression models. The source code is available at https://github.com/ktirta/regref.

2 Background

2.1 Combining Regression and Ranking

Huang et al. [2024] reformulate regression as joint regression–pairwise-ranking objectives and show that the combination improves predictive accuracy. Tynes et al. [2021], Fralish et al. [2023] instead train a regressor to predict pairwise differences in molecular properties; at inference, a query molecule is compared with reference molecules of known labels, and the mean of the predicted differences yields the final estimate. The post hoc method proposed by Gonçalves et al. [2023] refines regression outputs by computing a weighted average of the original prediction and those of its top-k nearest neighbors.

A closely related approach is the consolidation method for document relevance proposed by Yan et al. [2024]. Given a pointwise rating and a set of pairwise comparisons, their post-processor makes minimal adjustments to preserve the ranker-derived inequalities. For a one-dimensional rating $\hat{y}$ this projection simplifies to

$$\hat{y}^* = \max\big(\min(\hat{y}, U), L\big),$$

where L and U are the lower and upper bounds of the feasible interval.

Preference learning with Gaussian processes [Chu and Ghahramani, 2005, Houlsby et al., 2012, Chau et al., 2022] utilize pairwise data but differ fundamentally from our setting. Preference learning assumes a latent score $f(x)$ whose absolute scale is irrelevant; the aim is to preserve ordering and learn-to-rank, not to predict calibrated values. In contrast, we target real-world quantities, e.g., solubility or toxicity, and treat pairwise comparisons as auxiliary signals that refine supervised regression into meaningful, calibrated estimates. Similarly, many learning-to-rank approaches [Burges et al., 2005, Yildiz et al., 2020, Tom et al., 2024] may seem related to our work, but they actually address a different problem, i.e., ranking rather than regression.

2.2 LLM as a Pairwise Ranker

Large language models (LLMs) have recently been explored in molecular-property prediction and generation [Sun et al., 2025]. Several efforts fine-tune GPT-3- or LLaMA-based models with chemistry-specific prompts [Jablonka et al., 2024, Xie et al., 2024, Jacobs et al., 2024], while MolecularGPT [Liu et al., 2024] is instruction-tuned on 1 000 property-prediction tasks and achieves few-shot performance competitive with graph neural networks.

Fine-tuning, however, is resource-intensive and incurs substantial computational overhead. We therefore ask: can general-purpose, publicly hosted LLMs be leveraged directly, without domain-specific adaptation? We hypothesize that such models can supply useful signals when deployed as pairwise rankers.

Human cognition offers an analogue. People often judge quantities either by estimating absolute magnitudes (regression) or by deciding which of two items is greater (ranking). A long-standing hypothesis holds that comparative judgments are easier and less biased [Thurstone, 1927, Miller, 1956, Stewart et al., 2005]. Pairwise tasks impose lower cognitive load [Routh et al., 2023] and avoid scale-interpretation bias [Hoeijmakers et al., 2024]. Because LLMs are pretrained on corpora rich in relative statements and later aligned with human feedback [Zhang et al., 2023, Bai et al., 2022, Rafailov et al., 2023], they may inherit a similar advantage. As a real example, Guo et al. [2023]

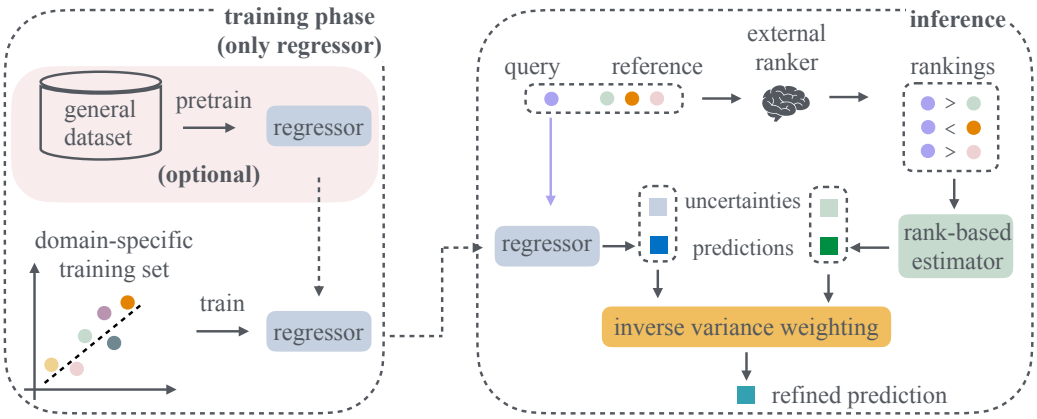


Figure 1: Overview of the `RankRefine` framework. During training, a regressor is trained using labeled data. At inference, a query sample is paired with reference samples with known properties and compared by an external ranker to produce pairwise rankings. These rankings are used to estimate the query's property value via a rank-based estimator. The final prediction is obtained by fusing the regressor's output and the rank-based estimate using inverse variance weighting.

show that off-the-shelf LLMs rival domain-specific models on chemistry classification and ranking benchmarks, underscoring the promise of pairwise prompting in specialized tasks.

3 Method

Our goal is to reduce the test-time error of an existing regressor by fusing its prediction with pairwise-ranking information in a plug-and-play fashion. We present `RankRefine`, a framework that augments a base regressor with rankings from an external source (e.g., an LLM or a domain expert). `RankRefine` is largely model-agnostic and applies to any regressor that returns both a point estimate and an uncertainty measure, for example Gaussian processes or random forests.

We assume access to a reference set $\mathbb{D} = \{(x_i, y_i)\}_{i=1}^{k}$ with known property values. This reference set can be drawn from the training set. At inference, the regressor predicts the property of a query sample x_0, giving $\hat{y}_0^{\text{reg}}$, while the ranker supplies pairwise comparisons between x_0 and each $x_i \in \mathbb{D}$.

3.1 `RankRefine`

Let $\hat{y}_0^{\text{reg}} \sim \mathcal{N}(y_0, \sigma_{\text{reg}}^2)$ be the regressor's unbiased Gaussian estimate of the true property y_0. A second, rank-based estimate $\hat{y}_0^{\text{rank}}$ is derived from the comparisons. Under these assumptions, the optimal fusion of the two estimates reduces to the inverse-variance weighting [Cochran and Carroll, 1953], formalized below.

Theorem 3.1 (RankRefine Fusion Theorem) *If $\hat{y}_0^{reg}$ and $\hat{y}_0^{rank}$ are independent, unbiased Gaussian estimators of y_0 with variances σ_{reg}^2 and σ_{rank}^2, the minimum-variance unbiased estimator is*

$$\hat{y}_0^{*} = \sigma_{post}^2 \left(\frac{\hat{y}_0^{reg}}{\sigma_{reg}^2} + \frac{\hat{y}_0^{rank}}{\sigma_{rank}^2} \right), \qquad \sigma_{post}^2 = \left(\frac{1}{\sigma_{reg}^2} + \frac{1}{\sigma_{rank}^2} \right)^{-1}. \tag{1}$$

See supplementary materials for the proof.

Rank-based estimate. Pairwise probabilities follow the Bradley–Terry model [Bradley and Terry, 1952], $P(x_i \succ x_j) = s(y_i - y_j)$ with sigmoid function $s(z) = \left(1 + e^{-z}\right)^{-1}$. Indexing the query as $i = 0$ and references as $i > 0$, we obtain $\hat{y}_0^{\text{rank}}$ by minimizing the negative log-likelihood

$$\hat{y}_0^{*\text{rank}} = \arg\min_{\hat{y}_0^{\text{rank}}} \Big[-\sum_{x_i \in \mathbb{A}} \log s(\hat{y}_0^{\text{rank}} - y_i) - \sum_{x_j \in \mathbb{B}} \log\big(1 - s(\hat{y}_0^{\text{rank}} - y_j)\big) \Big], \tag{2}$$

where $\mathbb{A}$ contains references ranked below x_0 and $\mathbb{B}$ those ranked above.

Lemma 3.2 (Variance of the rank-based estimate) *Let $\hat{y}_0^{*rank}$ minimize equation 2. Its variance is approximated by the inverse observed Fisher information [Ly et al., 2017],*

$$\sigma_{rank}^2 \approx \Big[\sum_{y_i \in \mathbb{A} \cup \mathbb{B}} s(\Delta_i)\big(1 - s(\Delta_i)\big) \Big]^{-1}, \quad \textit{with } \Delta_i = \hat{y}_0^{*rank} - y_i. \tag{3}$$

Applying Theorem 3.1 with σ_{rank}^2 from Lemma 3.2 yields $\hat{y}_0^*$.

3.2 Analysis of `RankRefine`

We measure performance via the mean absolute error (MAE). For a folded Gaussian derived from a zero-mean Gaussian, $\text{MAE} = \sqrt{2/\pi}\,\sigma$, so

$$\text{MAE}_{\text{post}} < \text{MAE}_{\text{reg}} \iff \sigma_{\text{post}}^2 < \sigma_{\text{reg}}^2.$$

Corollary 3.2.1 *Any informative ranker with finite variance ($\sigma_{rank}^2 < \infty$) lowers the expected MAE after fusion.*

More generally, letting $\alpha \in [0, 1]$ be the desired ratio between post-refinement and the original MAEs,

$$\text{MAE}_{\text{post}} \le \alpha\, \text{MAE}_{\text{reg}} \impliedby \sigma_{\text{rank}}^2 \le \frac{\alpha^2 \sigma_{\text{reg}}^2}{1 - \alpha^2}. \tag{4}$$

Regularization. If the ranker is biased yet over-confident ($\sigma_{\text{rank}}^2 \ll \sigma_{\text{reg}}^2$), we temper its variance via

$$\sigma_{\text{rank}}^2 \leftarrow \max\big(\sigma_{\text{rank}}^2,\, c\,\sigma_{\text{reg}}^2\big),$$

with user-chosen constant $c > 0$.

4 Experimental Results

We evaluate `RankRefine` on synthetic and real-world tasks. After outlining datasets, metrics, and implementation details, we report results in synthetic settings, where ranking oracle is available, and practical settings across multiple domains.

4.1 Experimental Setup

We use nine molecular datasets from the TDC ADME benchmark [Huang et al., 2021]: Caco-2 [Wang et al., 2016], Clearance Microsome and Clearance Hepatocyte [Di et al., 2012], log Half-Life [Obach et al., 2008], FreeSolv [Mobley and Guthrie, 2014], Lipophilicity [Wu et al., 2018], PPBR, Solubility [Sorkun et al., 2019], and VDss [Lombardo and Jing, 2016]. We additionally test three tabular regressions: crop-yield prediction from sensor data [Soundankar, 2025], student-performance prediction [Cortez, 2014], international-education cost estimation [Shamim, 2025]. In the human-as-ranker experiment, we use UTKFace [Zhang et al., 2017] for age estimation.

To emulate low-data regimes, we sample 50 training points from above datasets uniformly at random and merge the remainder with the original test split, repeating this re-split over five random seeds.

The primary metric is the mean absolute error (MAE). We report the *normalized error*

$$\beta = \frac{\text{MAE}_{\text{post}}}{\text{MAE}_{\text{reg}}},$$

where $\beta < 1$ indicates improvement.

Suppose that a pairwise ranker R evaluates a pair of inputs (x_i, x_j). The ranker returns $R(x_i, x_j) = 0$ if it predicts $x_i \succ x_j$, and $R(x_i, x_j) = 1$ otherwise. The pairwise ranker quality can be measured via pairwise ranking accuracy (PRA), defined as,

$$\text{PRA} = \frac{1}{|\mathbb{P}|} \sum_{(i,j) \in \mathbb{P}} \mathbb{I}\big(R(x_i, x_j) = \mathbb{I}(y_i < y_j)\big),$$

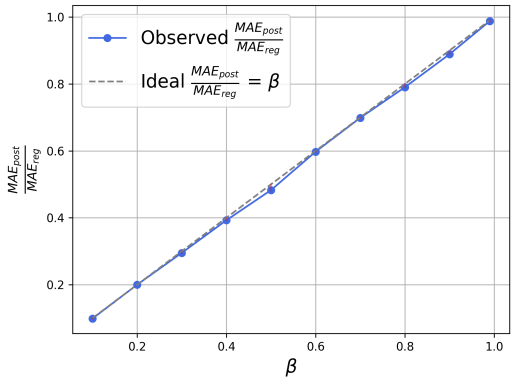


Figure 2: Empirical validation of the theoretical MAE reduction bound under ideal conditions. For each target improvement factor β, we compute the required ranker variance σ^2_{rank} using the right-side inequality in Implication 4. We simulate predictions from an oracle regressor ($\hat{y}_0^{\text{reg}} \sim \mathcal{N}(y_0, 1)$) and an oracle ranker ($\hat{y}_0^{\text{rank}} \sim \mathcal{N}(y_0, \sigma^2_{\text{rank}})$), and compute the fused estimate. The observed post-refinement MAE ratios match the ideal β values, demonstrating the correctness of the fusion rule under Gaussian distribution assumptions.

where $\mathbb{P} = \{(i, j)|i \neq j\}$ is the set of all comparable pairs with ground-truth labels y_i, y_j, and $\mathbb{I}$ is the indicator function which return 1 if the condition is true, and 0 otherwise. Unless stated otherwise, the base model is a random-forest regressor from scikit-learn [Pedregosa et al., 2011] with default hyper-parameters, executed on a single CPU.

4.2 Verifying Theoretical MAE Reduction Under Ideal Conditions

To validate Implication 4, we simulate unbiased oracle predictions with known variances. For target factors $\beta \in \{0.10, 0.20, \ldots, 0.90, 0.99\}$, the required ranker variance σ^2_{rank} is obtained from the right-side inequality of Implication 4. Synthetic estimates are drawn as $\hat{y}_0^{\text{reg}} \sim \mathcal{N}(y_0, 1)$ and $\hat{y}_0^{\text{rank}} \sim \mathcal{N}(y_0, \sigma^2_{\text{rank}})$. The fused prediction $\hat{y}_0^*$ follows Equation 1. Figure 2 confirms that empirical MAE ratios match the prescribed targets.

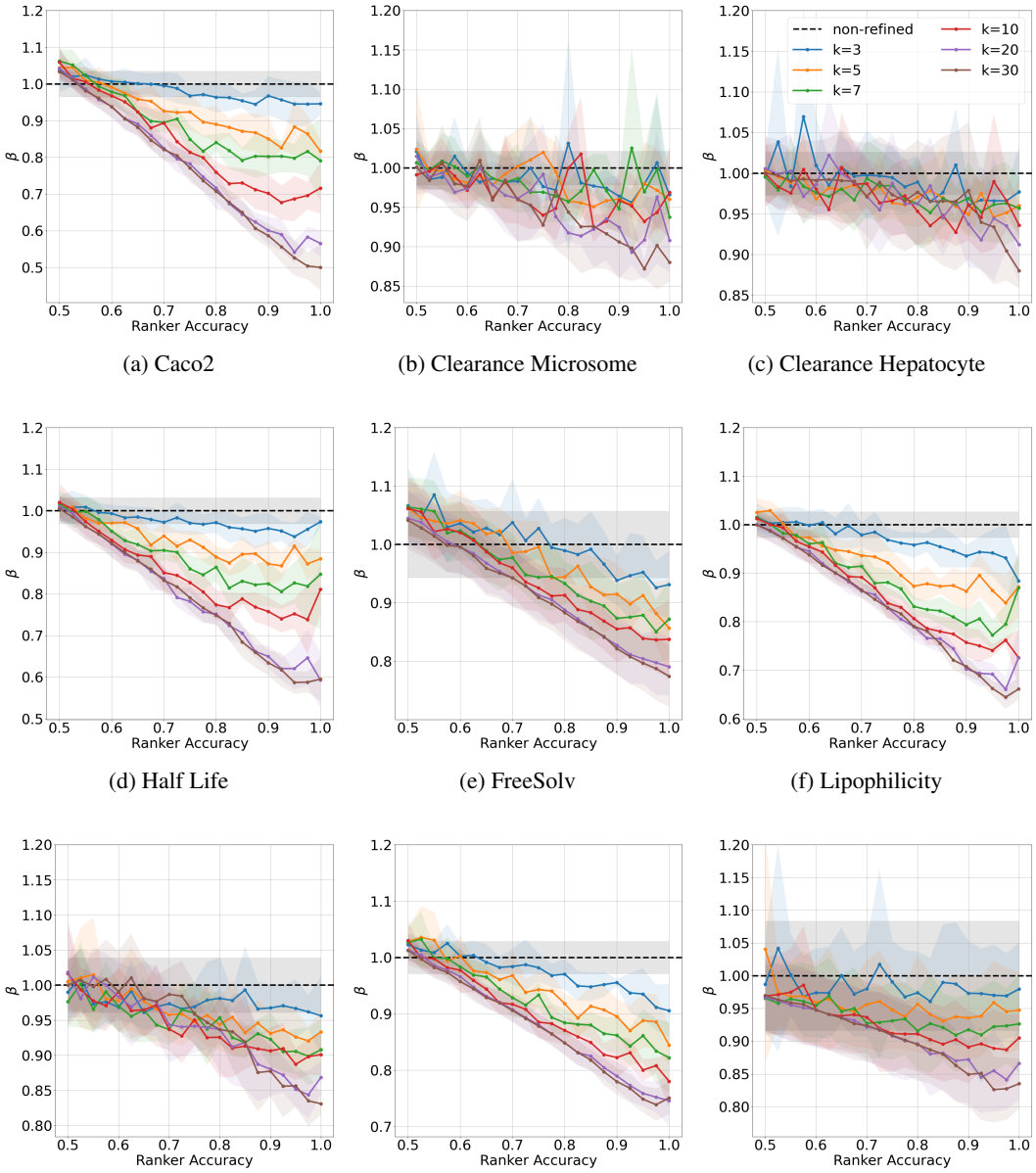
4.3 Effect of Ranker Accuracy and Reference Sample Size on Molecular-Property Prediction with `RankRefine`

Using the nine TDC datasets, we vary oracle ranker accuracy and the number k of reference comparisons per query. Figure 3 plots β, the post refinement MAE error divided by the regression error, versus pairwise ranker accuracy. `RankRefine` consistently lowers error, even with accuracy ≈ 0.55 and $k = 10$. Increasing the number of pairwise comparisons k tend to further lower β. We observe that $k = 20$ is optimal, as the difference between $k = 20$ and $k = 30$ is often minimal.

For very accurate rankers (> 0.95) a slight uptick in β arises from over-confident curvature estimates and extreme solutions when the query is an outlier. For the first source, recall that σ^2_{rank} is approximated by the second derivative of the negative log-likelihood. This curvature becomes smaller as the ranker accuracy approaches perfection, leading to over-confident estimates. For the second source, consider the case when the query is truly the smallest or largest value among the reference set. This is especially common when k is small and ranker is accurate. The optimizer may push $\hat{y}_0^{*\text{rank}}$ toward extreme values that still satisfy pairwise ranking constraints, but overshoot the true property value y_0.

4.4 Cross-Domain Generalization

Applying the same protocol to the three tabular datasets yields Figure 4, whose reductions mirror those in chemistry datasets. The results show that `RankRefine` generalizes beyond molecular tasks.

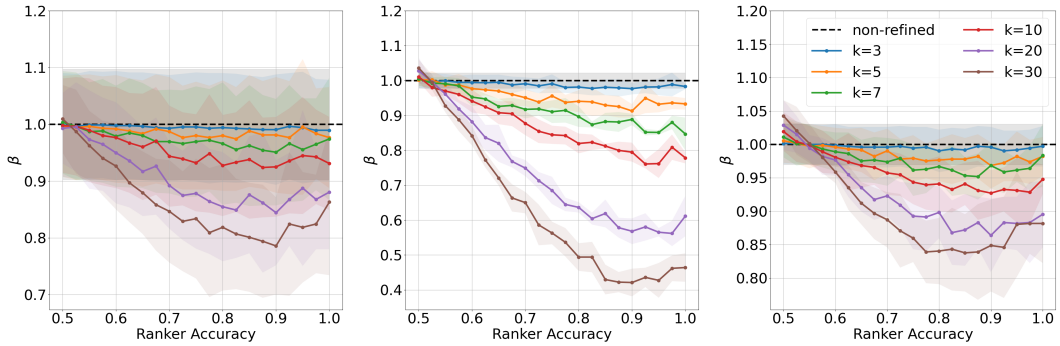


(a) Caco2 (b) Clearance Microsome (c) Clearance Hepatocyte

(d) Half Life (e) FreeSolv (f) Lipophilicity

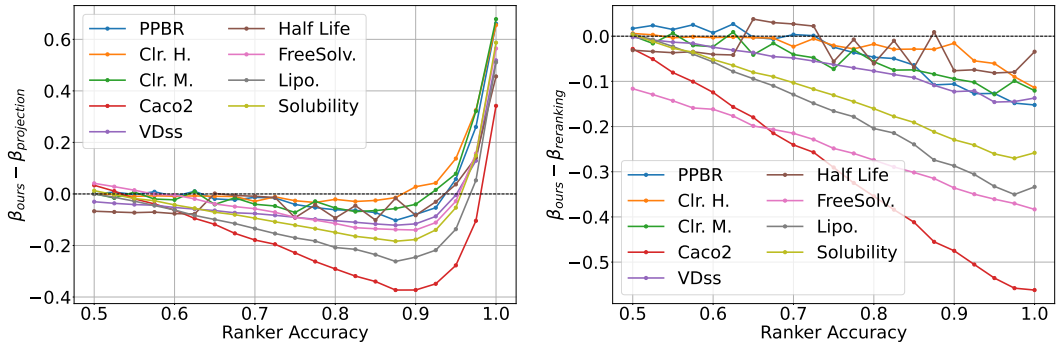
(g) PPBR (h) Solubility (i) VDss

Figure 3: Performance of `RankRefine` on molecular property prediction datasets under varying oracle ranker accuracy and number of reference comparisons. Each plot shows the normalized error $\beta = \frac{\text{MAE}_{\text{post}}}{\text{MAE}_{\text{reg}}}$ as a function of ranker accuracy, averaged over 5 random splits. $\beta < 1$ indicates improvements in regression performance over the base regressor. k is the number of pairwise comparisons for each test molecule. Shaded regions indicate standard deviation. Dashed gray line shows baseline MAE with no refinement. Across most configurations, `RankRefine` improves regression performance when using a ranker with accuracy as low as 0.55. Increasing k tends to lower β, but typically the benefits start to diminish beyond $k = 20$.



(a) International Education Cost (b) Smart Farming (c) Student Performance

Figure 4: Performance of `RankRefine` on tabular datasets under varying oracle ranker accuracy and number of reference comparisons. Across most configurations, `RankRefine` improves regression performance when using a ranker with accuracy as low as 0.55.



(a) (b)

Figure 5: (a) Comparison between `RankRefine` the constrained optimization (projection)-based refinement method of Yan et al. [2024] on molecular property prediction tasks. We report the difference in normalized error, $\beta_{\text{ours}} - \beta_{\text{projection}}$. A negative value indicates better performance by `RankRefine`. Each curve corresponds to a dataset. `RankRefine` generally outperforms the baseline when the ranker accuracy is between 0.5 and 0.95. (b) We also compare our method to a related post hoc regression improvement method, regression by re-ranking [Gonçalves et al., 2023].

4.5 Comparison with Other Baselines

We compare against two post hoc refinement methods: constrained optimization [Yan et al., 2024] and regression by re-ranking (RbR) [Gonçalves et al., 2023]. We reimplemented the two baselines from scratch, as no publicly available code was provided. We use $k = 30$ comparisons. Figure 5 displays $\beta_{\text{ours}} - \beta_{\text{projection}}$ and $\beta_{\text{ours}} - \beta_{\text{reranking}}$; negative values favour `RankRefine`. Our approach generally outperforms RbR in all datasets. Against the projection-based baseline, `RankRefine` excels when ranker accuracy lies between 0.50 and 0.95; projection dominates only when the ranker is nearly perfect. This is because, when all pairwise rankings are correct, the projection method strictly enforces the constraints without increasing the absolute error. In such cases, the MAE will either remain unchanged or improved, with the improvements depend primarily on the distribution and resolution of the known labels. Yet, such near-perfect rankers are rare in real-world scenarios. Thus, the robust performance of `RankRefine` with moderately-accurate rankers offers more practicality. More comparisons with projection-based refinement can be found in the supplementary materials.

4.6 Few-Shot Molecular-Property Prediction with LLMs

To evaluate the practical applicability of `RankRefine` in realistic few-shot settings, we replace the oracle ranker with ChatGPT-4o [OpenAI, 2025], a large language model (LLM). Note that we can

Table 1: PRA and resulting `RankRefine` performance using ChatGPT-4o as the ranker on six datasets from the TDC ADME benchmark. For each test molecule, 20 training molecules are randomly sampled to form pairwise comparisons. Values are averaged over 3 random splits. Despite moderate pairwise accuracy ($\sim$0.60–0.69), `RankRefine` consistently improves regression performance, demonstrating the viability of LLM-based rankers in low-data domains.

Dataset	Lipophilicity	Solubility	VDss
Pairwise Ranking Accuracy	0.622 ± 0.008	0.693 ± 0.035	0.605 ± 0.010
β	0.957 ± 0.012	0.934 ± 0.048	0.895 ± 0.053
Dataset	Caco2	Half Life	FreeSolv
Pairwise Ranking Accuracy	0.660 ± 0.013	0.602 ± 0.014	0.681 ± 0.050
β	0.970 ± 0.027	0.971 ± 0.005	0.937 ± 0.012

Table 2: Performance of `RankRefine` when using human participants as both direct estimators and pairwise rankers on a facial age estimation task. The base error (MAE_{reg}) corresponds to the raw human estimate error. Each test face is compared to 15 reference faces via human-judged pairwise comparisons. `RankRefine` improves prediction accuracy by refining each estimate based on human-provided rankings.

MAE_{reg}	Pairwise Ranking Acc.	β
6.343 ± 0.610	0.759 ± 0.052	0.954 ± 0.046

still evaluate the ranking accuracy of the LLM through oracle for the purpose of analysis. For each test molecule, we randomly sample 20 reference molecules from the training set and ask ChatGPT-4o to perform pairwise comparisons (e.g., "Which molecule is likely to have higher solubility?"). The model is queried using textual molecular descriptions, SMILES [Weininger, 1988]. The details of the prompt is provided in the supplementary materials.

Table 1 summarizes ChatGPT-4o ranking accuracy and the resulting β on six TDC datasets. Even moderate accuracies ($\approx$0.62–0.69) yield tangible MAE gains, underscoring both the robustness and practicality of our method.

One natural concern when employing a pretrained LLM as a ranker is whether it simply memorizes from training data which item (e.g., a molecule) has a higher or lower property value (e.g., solubility) compared to others. To test this, we conducted an additional evaluation on a private compound-activity dataset consisting of 75 compounds with IC50 labels measured against a specific target protein complex. This dataset is unpublished, and therefore, not part of any LLM training corpora, ruling out direct memorization. On this dataset, ChatGPT-4o achieves 60.14% pairwise ranking accuracy, indicating generalization capability and not simple memorization.

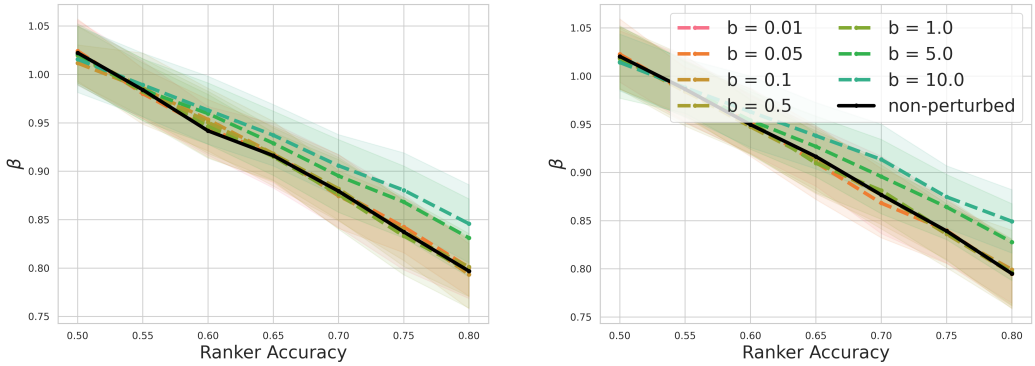
4.7 Human Age Estimation with Pairwise Self-Correction

`RankRefine` can potentially be used in an interactive, human-in-the-loop scenarios, where domain experts act as the pairwise rankers. As a proof of concept, we conduct a user study on the age estimation task. Details about the user study is available in the supplementary materials.

Table 2 reports a UTKFace user study where 6 participants are asked to: (1) estimate the age of 17 individuals from their corresponding pictures, and (2) perform pairwise judgments to 15 individuals in the reference set. Pairwise judgments raise ranking accuracy to 0.76 and cut MAE by $\approx$5 %, illustrating `RankRefine`'s value for expert self-correction.

4.8 Effects of Noisy Variance Estimates

We study the effect of noise in the ranker variance estimate on the Solubility dataset using $k = 30$ and $k = 40$. We add uniform perturbations sampled from [-b, +b] to the estimated ranker variance before fusion. As shown in Figure 6, performance degradation becomes noticeable when $b \geq 5$, which is more than three times the standard deviation of the ranker variance estimates. For reference, the mean and standard deviation of the estimated ranker variance are 2.358 ± 1.079.



(a) k = 30 (b) k = 40

Figure 6: Impact of noisy ranker variance estimate on the regression performance in Solubility dataset. Uniform noise from [-b, +b] is added to the estimated ranker variance. Performance starts to degrade when $b \geq 5$, more than 3 times the standard deviation of the ranker variance estimate.

4.9 Effects of Biased Regressor

We systematically inject constant bias into an otherwise unbiased regressor. Specifically, we uniformly draw ground-truth labels $y_0 \sim U(0, 1)$, and set predictions by injecting zero-mean Gaussian noise $\hat{y}_0^{reg} = y_0 + \epsilon$, such that MAE $\approx$ 1SD where SD is the standard deviation of the ground-truth labels. We then add a constant offset, B, increasing in magnitude from 0% to 70% of the label standard deviation (SD). We report the refined-to-original MAE ratio, β, for varying ranker accuracies. Table 3 in Appendix D shows that β worsens as bias increases, but remains $<$1 up to B = 60% SD, indicating that RankRefine remains effective even under substantial regressor bias.

4.10 Effects of Biased Sampling

We study the effects of non-uniform sampling of references, in particular, when the references are clustered. Similar to the biased regressor experiment, we generate synthetic predictions with ground-truth label range $[0, 1]$. We then restrict the reference labels to increasingly narrow subranges of the ground-truth label range, centered around the midpoint of the range. The reference set bias, $RB = x\%$, means that the reference labels cover only $(100 - x)\%$ of the full ground-truth range. For example, an RB = 10% means the reference labels range is [0.05, 0.95]. Table 4 in Appendix E suggests that RankRefine performance is best when the reference set spans the full ground-truth labels' range (i.e., RB = 0%), but clustered sets still yield substantial gains. For example, at RB = 90% and ranker accuracy = 60%, β = 0.8842, an 11.58% improvement.

Furthermore, we evaluate biased sampling in two challenging settings: (i) *disjoint*: reference labels are in [-1,0] or [1,2] while queries are in [0,1]; and (ii) *partial overlap*: reference labels are in [-0.5, 0.5] or [0.5, 1.5], while queries are in [0,1]. This simulates a distribution shift, where the distribution of the reference labels is different from the distribution of the query labels. Despite the challenging setups, RankRefine improves the base regressor when the ranker is moderately accurate (i.e., $\geq$65% in the disjoint case, $\geq$55% in the partial overlap case). More details are available in Table 5 in Appendix E.

5 Limitations

While `RankRefine` demonstrates strong empirical performance, it has a number of limitations:

- Our theoretical analysis assumes that both the regressor and rank-based estimates have unbiased, Gaussian-distributed errors and are independent. Despite the empirical robustness of our method, its assumptions may not hold in practice, particularly in the presence of heavy-tailed or skewed noise.
- `RankRefine` depends on well-calibrated uncertainty estimates from the regressor and a trustworthy estimate of ranker variance (via Equation 3). Miscalibration in either component can reduce or even negate the performance gains.
- Our oracle experiments assume uniformly random ranking errors; however, real-world rankers may exhibit systematic biases, for example consistently failing on extreme values which can skew the fusion process.
- `RankRefine` uses the Bradley–Terry model by treating true property values as proxies for latent scores, which differs from its original use where latent scores are learned to explain probabilistic pairwise outcomes. While practical, this introduces a modeling mismatch. A possible solution is to learn a mapping from a standard Bradley-Terry latent scores to labels on a holdout set and apply it during inference

6 Future Work

Future work can extend `RankRefine` in several promising directions. One avenue is to replace the Bradley-Terry likelihood with richer stochastic-transitivity models that better capture structured or systematic errors in the ranker. Another important direction involves incorporating Bayesian or conformal calibration layers to automatically correct misestimated variances and improve reliability. Beyond scalar predictions, `RankRefine` could potentially be extended to handle multivariate or structured targets, such as full pharmacokinetic profiles. Finally, leveraging rationale from language models could improve the interpretability of the ranking process and facilitate richer forms of expert feedback in decision-critical domains.

7 Conclusion

We introduced `RankRefine`, a plug-and-play, post hoc framework that injects pairwise-ranking signals into any uncertainty-aware regressor. Theoretically, we proved that inverse-variance fusion with a rank-based estimator lowers the expected mean absolute error (MAE) whenever the ranker variance is finite. Empirically, oracle simulations generally follows the theory, and experiments on nine molecular-property benchmarks plus three diverse tabular tasks showed consistent improvements even with ranker accuracies of around 55% and as few as 10 comparisons. ChatGPT-4o rankings yielded measurable MAE drops on six ADME datasets, and a user study demonstrated similar self-correction for human age estimation. Compared with a projection baseline, `RankRefine` prevailed whenever ranker accuracy lay in the realistic 0.50–0.95 range, underscoring its robustness and broad applicability.

References

Michael Sun, Minghao Guo, Weize Yuan, Veronika Thost, Crystal Elaine Owens, Aristotle Franklin Grosz, Sharvaa Selvan, Katelyn Zhou, Hassan Mohiuddin, Benjamin J Pedretti, et al. Representing molecules as random walks over interpretable grammars. In *International Conference on Machine Learning*, pages 46988–47016. PMLR, 2024.

Michael Sun, Gang Liu, Weize Yuan, Wojciech Matusik, and Jie Chen. Foundation molecular grammar: Multi-modal foundation models induce interpretable molecular graph languages. In *International Conference on Machine Learning*. PMLR, 2025.

Taicheng Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh Chawla, Olaf Wiest, Xiangliang Zhang, et al. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in Neural Information Processing Systems*, 36:59662–59688, 2023.

Pin-Yen Huang, Szu-Wei Fu, and Yu Tsao. Rankup: Boosting semi-supervised regression with an auxiliary ranking classifier. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=d2lPM1Aczs.

Michael Tynes, Wenhao Gao, Daniel J Burrill, Enrique R Batista, Danny Perez, Ping Yang, and Nicholas Lubbers. Pairwise difference regression: a machine learning meta-algorithm for improved prediction and uncertainty quantification in chemical search. *Journal of chemical information and modeling*, 61(8):3846–3857, 2021.

Zachary Fralish, Ashley Chen, Paul Skaluba, and Daniel Reker. Deepdelta: predicting admet improvements of molecular derivatives with deep learning. *Journal of cheminformatics*, 15(1):101, 2023.

Filipe Marcel Fernandes Gonçalves, Daniel Carlos Guimarães Pedronette, and Ricardo da Silva Torres. Regression by re-ranking. *Pattern Recognition*, 140:109577, 2023.

Le Yan, Zhen Qin, Honglei Zhuang, Rolf Jagerman, Xuanhui Wang, Michael Bendersky, and Harrie Oosterhuis. Consolidating ranking and relevance predictions of large language models through post-processing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 410–423, 2024.

Wei Chu and Zoubin Ghahramani. Preference learning with gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pages 137–144, 2005.

Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Jose Hernández-lobato. Collaborative gaussian processes for preference learning. *Advances in neural information processing systems*, 25, 2012.

Siu Lun Chau, Javier Gonzalez, and Dino Sejdinovic. Learning inconsistent preferences with gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 2266–2281. PMLR, 2022.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96, 2005.

Ilkay Yildiz, Jennifer Dy, Deniz Erdogmus, Jayashree Kalpathy-Cramer, Susan Ostmo, J Peter Campbell, Michael F Chiang, and Stratis Ioannidis. Fast and accurate ranking regression. In *International Conference on Artificial Intelligence and Statistics*, pages 77–88. PMLR, 2020.

Gary Tom, Stanley Lo, Samantha Corapi, Alan Aspuru-Guzik, and Benjamin Sanchez-Lengeling. Ranking over regression for bayesian optimization and molecule selection. *arXiv preprint arXiv:2410.09290*, 2024.

Kevin Maik Jablonka, Philippe Schwaller, Andres Ortega-Guerrero, and Berend Smit. Leveraging large language models for predictive chemistry. *Nature Machine Intelligence*, 6(2):161–169, 2024.

Zikai Xie, Xenophon Evangelopoulos, Ömer H Omar, Alessandro Troisi, Andrew I Cooper, and Linjiang Chen. Fine-tuning gpt-3 for machine learning electronic and functional properties of organic molecules. *Chemical science*, 15(2):500–510, 2024.

Ryan Jacobs, Maciej P Polak, Lane E Schultz, Hamed Mahdavi, Vasant Honavar, and Dane Morgan. Regression with large language models for materials and molecular property prediction. *arXiv preprint arXiv:2409.06080*, 2024.

Yuyan Liu, Sirui Ding, Sheng Zhou, Wenqi Fan, and Qiaoyu Tan. Moleculargpt: Open large language model (llm) for few-shot molecular property prediction. *arXiv preprint arXiv:2406.12950*, 2024.

LL Thurstone. A law of comparative judgment. *Psychological Review*, 34(4), 1927.

George A Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

Neil Stewart, Gordon DA Brown, and Nick Chater. Absolute identification by relative judgment. *Psychological review*, 112(4):881, 2005.

Jennifer Routh, Sharmini Julita Paramasivam, Peter Cockcroft, Sarah Wood, John Remnant, Cornélie Westermann, Alison Reid, Patricia Pawson, Sheena Warman, Vishna Devi Nadarajah, et al. Rating and ranking preparedness characteristics important for veterinary workplace clinical training: a novel application of pairwise comparisons and the elo algorithm. *Frontiers in Medicine*, 10: 1128058, 2023.

Eva JI Hoeijmakers, Bibi Martens, Babs MF Hendriks, Casper Mihl, Razvan L Miclea, Walter H Backes, Joachim E Wildberger, Frank M Zijta, Hester A Gietema, Patricia J Nelemans, et al. How subjective ct image quality assessment becomes surprisingly reliable: pairwise comparisons instead of likert scale. *European Radiology*, 34(7):4494–4503, 2024.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

William G Cochran and Sarah Porter Carroll. A sampling investigation of the efficiency of weighting inversely as the estimated variance. *Biometrics*, 9(4):447–459, 1953.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Alexander Ly, Maarten Marsman, Josine Verhagen, Raoul PPP Grasman, and Eric-Jan Wagenmakers. A tutorial on fisher information. *Journal of Mathematical Psychology*, 80:40–55, 2017.

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *Advances in neural information processing systems*, 2021.

Ning-Ning Wang, Jie Dong, Yin-Hua Deng, Min-Feng Zhu, Ming Wen, Zhi-Jiang Yao, Ai-Ping Lu, Jian-Bing Wang, and Dong-Sheng Cao. Adme properties evaluation in drug discovery: prediction of caco-2 cell permeability using a combination of nsga-ii and boosting. *Journal of chemical information and modeling*, 56(4):763–773, 2016.

Li Di, Christopher Keefer, Dennis O Scott, Timothy J Strelevitz, George Chang, Yi-An Bi, Yurong Lai, Jonathon Duckworth, Katherine Fenner, Matthew D Troutman, et al. Mechanistic insights from comparing intrinsic clearance values between human liver microsomes and hepatocytes to guide drug design. *European journal of medicinal chemistry*, 57:441–448, 2012.

R Scott Obach, Franco Lombardo, and Nigel J Waters. Trend analysis of a database of intravenous pharmacokinetic parameters in humans for 670 drug compounds. *Drug Metabolism and Disposition*, 36(7):1385–1405, 2008.

David L Mobley and J Peter Guthrie. Freesolv: a database of experimental and calculated hydration free energies, with input files. *Journal of computer-aided molecular design*, 28:711–720, 2014.

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.

Murat Cihan Sorkun, Abhishek Khetan, and Süleyman Er. Aqsoldb, a curated reference set of aqueous solubility and 2d descriptors for a diverse set of compounds. *Scientific data*, 6(1):143, 2019.

Franco Lombardo and Yankang Jing. In silico prediction of volume of distribution in humans. extensive data set and the exploration of linear and nonlinear methods coupled with molecular interaction fields descriptors. *Journal of chemical information and modeling*, 56(10):2042–2052, 2016.

Atharva Soundankar. Smart farming sensor data for yield prediction, 2025. URL `https://www.kaggle.com/datasets/atharvasoundankar/smart-farming-sensor-data-for-yield-prediction`.

Paulo Cortez. Student Performance. UCI Machine Learning Repository, 2014.

Adil Shamim. Cost of international education, 2025. URL `https://www.kaggle.com/datasets/adilshamim8/cost-of-international-education`.

Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5810–5818, 2017.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

OpenAI. Chatgpt-4o, 2025. URL `https://chatgpt.com/`.

David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction accurately reflects the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in a separate section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.

- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Assumptions and proofs are provided in the Method section.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Implementation details are provided in the Experimental Results section and Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.

- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code will be shared to the public upon publication.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Training and test details are described in the Experimental Results section and Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The experimental results are supplemented with appropriate statistical information.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Computer resources information is described in the Experimental Results section.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This work studies a generic post-hoc method for improving the performance of a regressor using pairwise ranking predictions.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There is no high risk for misuse associated with this method.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Original owners of assets are property credited in the reference.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The public code will be accompanied by documentation.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: Full text of instructions given to participants and screenshots are provided in the supplementary materials.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: There are no potential risks incurred by study participants.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The usage of LLMs is described on multiple occasions in the manuscript and supplementary materials.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Extended Proof

A.1 `RankRefine` is a minimum-variance unbiased estimator

Suppose that we have two independent unbiased estimates of the same parameter y_0,

$$\hat{y}_0^{\text{reg}} \sim \mathcal{N}(y_0, \sigma_{\text{reg}}^2), \qquad \hat{y}_0^{\text{rank}} \sim \mathcal{N}(y_0, \sigma_{\text{rank}}^2),$$

and we want to form a linear combination of the two estimates,

$$\hat{y}_0^* = w\hat{y}_0^{\text{reg}} + (1-w)\hat{y}_0^{\text{rank}}, \tag{5}$$

our goal is to choose $w \in [0, 1]$ such that the variance is minimized.

For two independent random variables X and Y, $\text{Var}(aX + bY) = a^2X + b^2Y$. The post-fusion variance of the unbiased estimates is then,

$$\sigma_{\text{post}}^2 = w^2\sigma_{\text{reg}}^2 + (1-w)^2\sigma_{\text{rank}}^2. \tag{6}$$

To minimize σ_{post}^2 with respect to w, we take the derivative and solve for,

$$\begin{aligned}
\frac{d}{dw}\left(w^2\sigma_{\text{reg}}^2 + (1-w)^2\sigma_{\text{rank}}^2\right) &= 0, \\
2w\sigma_{\text{reg}}^2 - 2(1-w)\sigma_{\text{rank}}^2 &= 0 \\
2w\sigma_{\text{reg}}^2 &= 2(1-w)\sigma_{\text{rank}}^2 \\
w\sigma_{\text{reg}}^2 &= (1-w)\sigma_{\text{rank}}^2 \\
w\sigma_{\text{reg}}^2 &= \sigma_{\text{rank}}^2 - w\sigma_{\text{rank}}^2 \\
w\sigma_{\text{reg}}^2 + w\sigma_{\text{rank}}^2 &= \sigma_{\text{rank}}^2 \\
w(\sigma_{\text{reg}}^2 + \sigma_{\text{rank}}^2) &= \sigma_{\text{rank}}^2.
\end{aligned} \tag{7}$$

Therefore,

$$w = \frac{\sigma_{\text{rank}}^2}{\sigma_{\text{reg}}^2 + \sigma_{\text{rank}}^2}, \qquad (1-w) = \frac{\sigma_{\text{reg}}^2}{\sigma_{\text{reg}}^2 + \sigma_{\text{rank}}^2}. \tag{8}$$

Substituting the weights in Equation 6 with Equation 8, we obtain the `RankRefine` post-fusion variance in Equation 1,

$$\begin{aligned}
\sigma_{\text{post}}^2 &= \left(\frac{\sigma_{\text{rank}}^2}{\sigma_{\text{reg}}^2 + \sigma_{\text{rank}}^2}\right)^2 \sigma_{\text{reg}}^2 + \left(\frac{\sigma_{\text{reg}}^2}{\sigma_{\text{reg}}^2 + \sigma_{\text{rank}}^2}\right)^2 \sigma_{\text{rank}}^2 \\
&= \frac{(\sigma_{\text{rank}}^2)^2\sigma_{\text{reg}}^2 + (\sigma_{\text{reg}}^2)^2\sigma_{\text{rank}}^2}{(\sigma_{\text{reg}}^2 + \sigma_{\text{rank}}^2)^2} \\
&= \frac{\sigma_{\text{rank}}^2\sigma_{\text{reg}}^2(\sigma_{\text{rank}}^2 + \sigma_{\text{reg}}^2)}{(\sigma_{\text{reg}}^2 + \sigma_{\text{rank}}^2)^2} \\
&= \frac{\sigma_{\text{rank}}^2\sigma_{\text{reg}}^2}{\sigma_{\text{rank}}^2 + \sigma_{\text{reg}}^2} \\
&= \left(\frac{1}{\sigma_{\text{reg}}^2} + \frac{1}{\sigma_{\text{rank}}^2}\right)^{-1}
\end{aligned} \tag{9}$$

Similarly, substituting the weights in Equation 5 with Equation 8, we get the `RankRefine` estimate in Equation 1,

$$\begin{aligned}
\hat{y}_0^* &= \frac{\sigma^2_{\text{rank}}}{\sigma^2_{\text{reg}} + \sigma^2_{\text{rank}}}\, \hat{y}_0^{\text{reg}} + \frac{\sigma^2_{\text{reg}}}{\sigma^2_{\text{reg}} + \sigma^2_{\text{rank}}}\, \hat{y}_0^{\text{rank}} \\
&= \frac{\sigma^2_{\text{rank}}\sigma^2_{\text{reg}}}{(\sigma^2_{\text{reg}} + \sigma^2_{\text{rank}})\sigma^2_{\text{reg}}} \hat{y}_0^{\text{reg}} + \frac{\sigma^2_{\text{reg}}\sigma^2_{\text{rank}}}{(\sigma^2_{\text{reg}} + \sigma^2_{\text{rank}})\sigma^2_{\text{rank}}} \hat{y}_0^{\text{rank}} \\
&= \frac{\sigma^2_{\text{post}}}{\sigma^2_{\text{reg}}} \hat{y}_0^{\text{reg}} + \frac{\sigma^2_{\text{post}}}{\sigma^2_{\text{rank}}} \hat{y}_0^{\text{rank}} \\
&= \sigma^2_{\text{post}} \left(\frac{\hat{y}_0^{\text{reg}}}{\sigma^2_{\text{reg}}} + \frac{\hat{y}_0^{\text{rank}}}{\sigma^2_{\text{rank}}} \right)
\end{aligned} \tag{10}$$

A.2 Analysis of `RankRefine`

Corollary 3.2.1 is obtained from:

$$\begin{aligned}
\sigma^2_{\text{post}} < \sigma^2_{\text{reg}} &\iff \left(\frac{1}{\sigma^2_{\text{reg}}} + \frac{1}{\sigma^2_{\text{rank}}} \right)^{-1} < \sigma^2_{\text{reg}}, \\
&\iff \frac{1}{\sigma^2_{\text{reg}}} + \frac{1}{\sigma^2_{\text{rank}}} > \frac{1}{\sigma^2_{\text{reg}}}, \\
&\iff \frac{1}{\sigma^2_{\text{rank}}} < 0, \\
&\iff \sigma^2_{\text{rank}} < \infty.
\end{aligned} \tag{11}$$

For an unbiased Gaussian estimates $\hat{y}_0 \sim \mathcal{N}(y_0, \sigma^2)$, the expected error is,

$$\begin{aligned}
\mathbb{E}(|\hat{y}_0 - y_0|) &= \mathbb{E}(|y_0 + \epsilon - y_0|), \qquad \text{with } \epsilon \sim \mathcal{N}(0, \sigma^2) \\
&= \mathbb{E}(|\epsilon|).
\end{aligned} \tag{12}$$

Since ϵ is a Gaussian distribution, then $|\epsilon|$ follows the half-Gaussian distribution with an expected value equals to $\sqrt{2/\pi}\sigma$. The subsequent β bound in Equation 4 is obtained from:

$$\begin{aligned}
\text{MAE}_{\text{post}} \leq \beta\ \text{MAE}_{\text{reg}} &\iff \sigma_{\text{post}} \leq \beta\ \sigma_{\text{reg}}, \\
&\iff \sigma^2_{\text{post}} \leq \beta^2\ \sigma^2_{\text{reg}}, \\
&\iff \left(\frac{1}{\sigma^2_{\text{reg}}} + \frac{1}{\sigma^2_{\text{rank}}} \right)^{-1} \leq \beta^2\ \sigma^2_{\text{reg}} \\
&\iff \frac{1}{\sigma^2_{\text{reg}}} + \frac{1}{\sigma^2_{\text{rank}}} \geq \frac{1}{\beta^2\ \sigma^2_{\text{reg}}} \\
&\iff \frac{1}{\sigma^2_{\text{rank}}} \geq \frac{1 - \beta^2}{\beta^2\ \sigma^2_{\text{reg}}}, \\
&\iff \sigma^2_{\text{rank}} \leq \frac{\beta^2\ \sigma^2_{\text{reg}}}{1 - \beta^2}
\end{aligned} \tag{13}$$

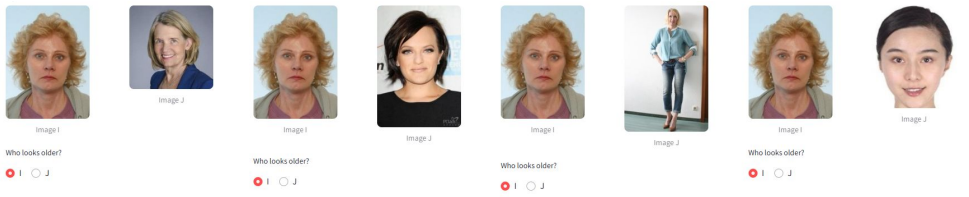
B Additional Experimental Details

B.1 Prompt for obtaining pairwise rankings from ChatGPT-4o

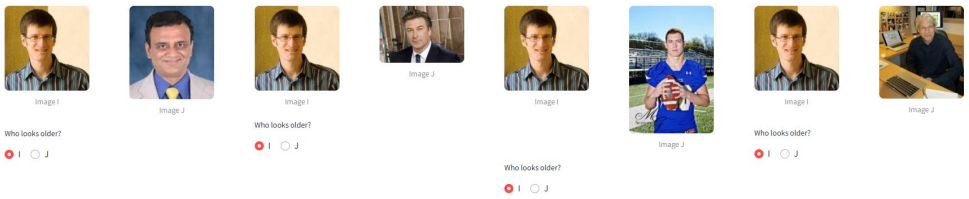
We use the following prompt to query ChatGPT-4o to predict the pairwise rankings for the Lipophilicity dataset.

```
# Identity
```

```
You are an expert in chemistry and biology. Given a short description of a
molecular property and two molecular SMILES, you can determine if Molecule A
has greater property value than Molecule B or not.

# Instructions

* The list of pairwise molecules is given in two CSV files: "test_A.csv" and
"test_B.csv". These are two different lists that you need to perform rank
prediction on. The first line is the header.
* You cannot use external cheminformatics library to directly predict the property.
* You can design your own heuristics, comparing atom types, bonds,
and other important information to make your predictions.
* Before you start designing the heuristics, briefly explain the property
and what influence its values. Incorporate this prior knowledge into your heuristic.
* You are given 5 examples in the prompt. Make sure your heuristics in
general is aligned with the given examples. Pay attention to the examples,
especially because some properties have greater effects when its values are lower.
* You should output CSV files titled "test_pred_A.csv" and "test_pred_B".
The header is "molecule_a, molecule_b, is_a_greater".
* Only output 0 or 1 for "is_a_greater". 0 means that the property value of
molecule A is less than that of Molecule B. 1 means that the property value of
molecule A is greater than that of Molecule B
* In addition to the output CSV file, you should output the first and last
molecule pairs, in both CSV A and CSV B, to the chat. Briefly explain your
answer for these three molecule pairs.

# Examples

<user_query>
The property of interest is lipophilicity, the ability of a drug to dissolve in a
lipid (e.g. fats, oils) environment.
Examples:
CC(C)Cn1c(=O)n(C)c(=O)c2c(C(=O)N3CC[C@@H](O)C3)c(Oc3cccc4ccccc34)sc21,
CC(C)(C(=O)O)c1ccc(C(O)CCCN2CCC(C(O)(c3ccccc3)c3ccccc3)CC2)cc1,
1
O=C(Nc1ccccc1Cl)c1cc[nH]n1,
O=C(Nc1ccc(Nc2ccc(NC(=O)c3ccccc3)c3c2C(=O)c2ccccc2C3=O)c2c1C(=O)c1ccccc1C2=O)c1ccccc1,
0
Cn1c2ccccc2c2cc(NC(=O)CCc3ccncc3)ccc21,
Cc1cc(OCCCS(C)(=O)=O)cc(C)c1-c1cccc(COc2ccc3c(c2)OC[C@H]3CC(=O)O)c1,
1
Cc1ccnc2nc(C(=O)Nc3nccs3)nn12,
Cc1cc(OCCCS(C)(=O)=O)cc(C)c1-c1cccc(COc2ccc3c(c2)OC[C@H]3CC(=O)O)c1,
0
O=C(NCc1ccc(OC(F)(F)F)cc1)C1c2ccccc2C(=O)N1CC1CC(F)(F)C1,
COCCCOc1ccnc(C[S+]([O-])c2nc3ccccc3[nH]2)c1C,
1
</user_query>
```



(a)



(b)

Figure 7: Partial interface used in the user study. Participants first estimate the age of each test individual, then make pairwise judgments on who appears older between the test individual and reference individuals of the same UTKFace gender label.

B.2 User Study

The user study uses age estimation on the UTKFace dataset as a regression task. Six participants estimate the ages of 17 individuals from facial images. Each test individual is then paired with 15 disjoint reference individuals, and participants judge who appears older in each pair based solely on facial appearance. To prevent fatigue, the total number of comparisons is limited to $17 \times 15 = 255$. A screenshot of the Streamlit app used in the study is shown in Figure 7.

C More Comparisons with the Projection-based Approach

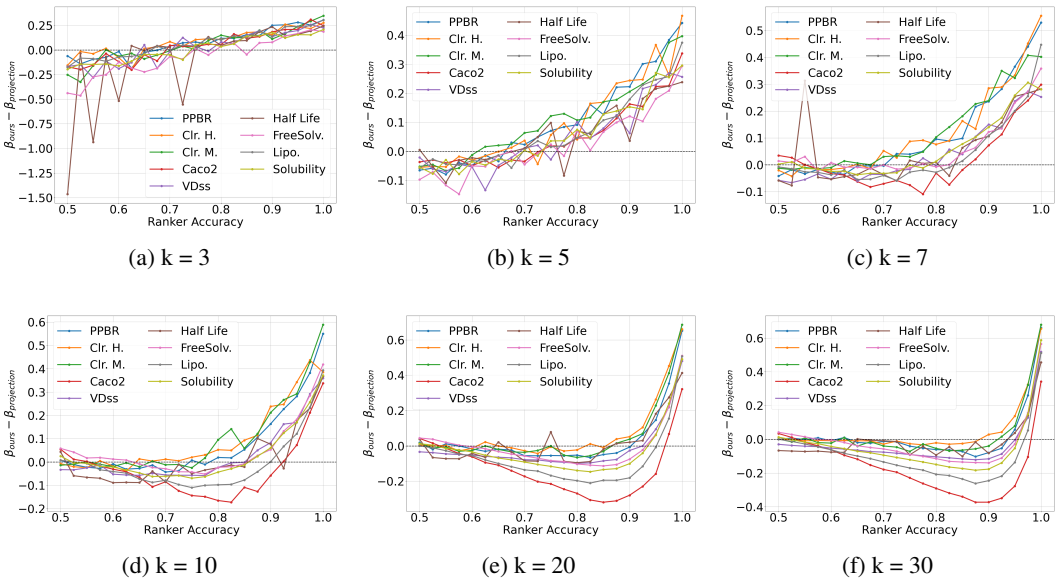
Figure 8 highlights how performance varies between `RankRefine` and the projection-based approach as the number of reference samples increases. While the projection-based method benefits from high-accuracy rankers or minimal references, `RankRefine` demonstrates stronger robustness in lower-accuracy regimes, particularly as the number of reference samples k increases.

D Ablation: Biased Regressor

We inject constant bias B into an otherwise unbiased regressor, where the magnitude of the offset is measured as a percentage of the ground truth labels standard deviation. RankRefine offers improvement even when the regressor is biased (e.g., 60% of the SD of the ground truth labels)

Table 3: RankRefine performance, measured in β (lower is better), on a biased regressor.

Ranker Accuracy	**50%**	**55%**	**60%**	**65%**	**70%**
Bias (B) = 0% SD	0.8911	0.8436	0.8694	0.8668	0.8949
B = 10% SD	0.8995	0.8800	0.8623	0.8602	0.8969
B = 20% SD	0.9084	0.8869	0.8704	0.8704	0.9047
B = 30% SD	0.9224	0.8997	0.8844	0.8861	0.9192
B = 40% SD	0.9409	0.9172	0.9059	0.9071	0.9395
B = 50% SD	0.9645	0.9403	0.9329	0.9343	0.9654
B = 60% SD	0.9942	0.9702	0.9649	0.9670	0.9967
B = 70% SD	1.0291	1.0062	1.0022	1.0049	1.0326



(a) k = 3 (b) k = 5 (c) k = 7

(d) k = 10 (e) k = 20 (f) k = 30

Figure 8: Comparison of `RankRefine` and the projection-based approach across varying reference sample sizes k. The projection-based method performs better with high ranker accuracy or very few references, while `RankRefine` excels when accuracy is below 70% for $k = 7$, 80% for $k = 10$, 90% for $k = 20$, and 95% for $k = 30$. Since `RankRefine` can leverage general-purpose LLMs, collecting more pairwise rankings can be done efficiently as long as the reference set is sufficiently large.

E Ablation: Biased Sampling

We restrict the reference set range to increasingly narrow subsets of the ground truth label range ([0, 1]), simulating clustered or biased samples. The reference set bias, $RB = x\%$, means that the reference labels cover only $(100 - x)\%$ of the full ground-truth range. For example, an RB = 10% means the reference labels range is [0.05, 0.95]. Best performance is achieved when the reference set spans the entire range of the ground truth labels, but clustered reference sets still provide substantial regression improvements.

Table 4: RankRefine performance, measured in β, with non-uniform sampling of the reference set.

Ranker Accuracy	**50%**	**55%**	**60%**	**65%**	**70%**
Reference Set Bias (RB) = 0% range	0.8828	0.8706	0.8343	0.8443	0.8598
RB = 10% range	0.9034	0.8731	0.8493	0.8529	0.8768
RB = 20% range	0.9078	0.8753	0.8366	0.8694	0.9002
RB = 30% range	0.9076	0.8701	0.8439	0.8733	0.9280
RB = 40% range	0.9143	0.8767	0.8618	0.8824	0.9414
RB = 50% range	0.9253	0.8786	0.8640	0.8769	0.9480
RB = 60% range	0.9349	0.8858	0.8712	0.8929	0.9546
RB = 70% range	0.9385	0.8919	0.8770	0.8957	0.9597
RB = 80% range	0.9551	0.8984	0.8798	0.8943	0.9558
RB = 90% range	0.9664	0.9042	0.8842	0.9035	0.9631

To test the impact of distribution shift between reference and query, we created two challenging scenarios where the reference labels and query labels are drawn uniformly from different range values:

- *Disjoint* distributions: reference labels in [-1, 0] or [1, 2], query labels in [0, 1]. We averaged the results of the two cases, that is, between ref = [-1, 0], query = [0, 1] and ref = [1, 2], query = [0, 1].

- *Partial overlap*: reference labels in [-0.5, 0.5] or [0.5, 1.5], query labels in [0, 1]. We averaged the results of the two cases, that is, between ref = [-0.5, 0.5], query = [0, 1] and ref = [0.5, 1.5], query = [0, 1].

The results in 5 shows that RankRefine still improves the base regressor when the ranker is moderately accurate, even when the distribution of the reference labels is different from the query.

Table 5: RankRefine performance, measured in β, for disjoint and partially overlapping reference set.

Ranker Accuracy	**50%**	**55%**	**60%**	**65%**	**70%**
Disjoint	1.2906	1.1629	1.0170	0.9437	0.9219
Partial Overlap	1.0160	0.9465	0.8700	0.8607	0.8898