RELIABILITY-CALIBRATED POSE-BY-METRIC FUSION FOR TARGET-LOCAL VIRTUAL SCREENING

Anonymous authors
Paper under double-blind review

ABSTRACT

Structure-based virtual screening ranks candidate ligands against a target protein by aggregating heterogeneous evidence from predicted protein–ligand complexes, structural confidence, docking affinity, interface geometry, interaction fingerprints, and ligand descriptors, yet within-target ranking quality remains decisive for downstream hit rates. Two difficulties complicate this aggregation: binding is pose-dependent with uncertain pose quality at inference time, and supervised ranking signals are sharply non-uniform across targets, with metric reliability also varying per target. We propose **PoT-Rank** (**Po**se-**T**rust Ranking), a target-local learning-to-rank framework that represents each candidate as a masked pose-by-metric tensor and learns hierarchical softmax weights over feature suites, metrics, and poses. Metric trust is calibrated without label leakage through a training-only reliability gate based on same-target pairwise concordance, paired with a target-normalized hard-decoy curriculum. To address differing statistical risks, PoT-Rank splits into a dense pose-aware ranker with a target-balanced pairwise hinge loss and a sparse antisymmetric comparator with signed-vote synthesis, with all stateful components frozen before evaluation. Under a unified frozen protocol with protein-side MMseqs2 splits at 40% identity, PoT-Rank attains the strongest within-target ranking in both regimes, reaching within-target pairwise accuracy 0.6602 and Spearman 0.5343 on the dense split (3 targets, 565 within-target pairs) and 0.6897 and 0.4313 on the sparse split (79 targets). It surpasses classical scoring functions, deep affinity predictors, graph-based models, pose-aware rescorers, and structure-prediction-based affinity models, and matches or exceeds strong tabular baselines in both regimes, while remaining stable under leakage audits, multi-seed perturbation, and corrupted-channel stress tests. Our code is available at `https://anonymous.4open.science/r/PoT-Rank-4CD1/`.

1 INTRODUCTION

Drug discovery hinges on identifying small-molecule ligands that bind a target protein with affinity and specificity to modulate its function, and the geometric and energetic complementarity of the resulting protein–ligand complex is what ultimately determines binding (Shoichet, 2004); structure-based virtual screening operationalizes this view by docking each candidate ligand into the target binding pocket, scoring the resulting complex, and prioritizing which ligands enter expensive experimental validation, so that within-target ranking quality directly governs the hit rate of the downstream pipeline (Bender et al., 2021; Shoichet, 2004). Modern screening stacks no longer rely on a single physics-inspired score, but instead aggregate heterogeneous signals such as structural-confidence estimates from co-folding models (Jumper et al., 2021; Wohlwend et al., 2025), docking affinities (Koes et al., 2013; Trott & Olson, 2010), interface-geometry descriptors (McGibbon et al., 2015; Michaud-Agrawal et al., 2011), interaction fingerprints (Bouysset & Fiorucci, 2021), and ligand-level chemical properties[1]. The central question is not which score is best in isolation, but how these signals should be combined so that, for a given target, stronger binders are reliably ranked above weaker ones.

Two structural difficulties complicate this aggregation. First, protein–ligand binding is intrinsically pose-dependent: a flexible ligand can dock into the target pocket in multiple plausible orientations

[1] `https://www.rdkit.org/`

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

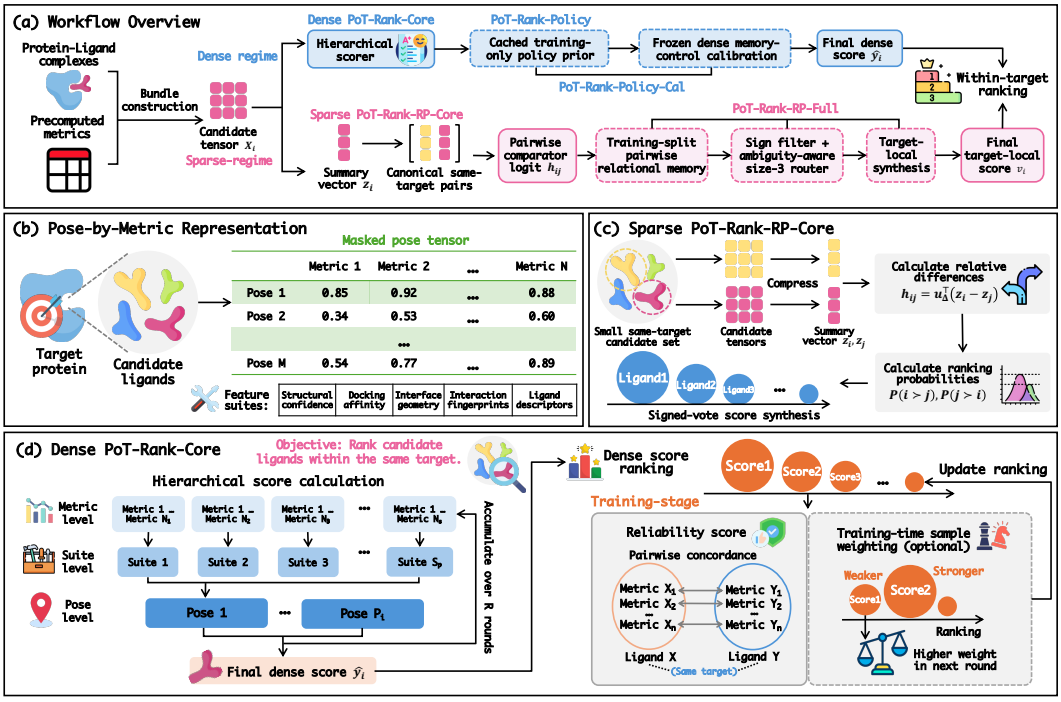


Figure 1: **Overview of PoT-Rank.** **(a)** Workflow: each protein–ligand complex is bundled into a candidate tensor $\mathbf{X}_i$ and routed to the dense or sparse branch by a fixed manifest-level regime assignment; all stateful components are frozen before test evaluation. **(b)** Pose-by-metric representation: $P_{\max}$=10 poses by Q=13 channels, partitioned into five protein–ligand evidence suites. **(c)** Sparse PoT-Rank-RP-Core: candidate summaries form an antisymmetric pair logit $h_{ij} = \mathbf{u}_\Delta^\top(\mathbf{z}_i - \mathbf{z}_j)$, turned into target-local rankings by signed-vote synthesis. **(d)** Dense PoT-Rank-Core: a hierarchical scorer fuses pose features through suite-, metric-, and pose-level softmax weights over R_f rounds, with a training-only reliability gate and a hard-decoy curriculum.

and conformations, and the binding free energy depends on which residue-level contacts, hydrogen bonds, hydrophobic packing, π-stacking, are actually realized in that pose, yet pose quality is uncertain at inference time (Corso et al., 2024; 2022), so naive pose averaging dilutes informative signals while top-1 selection discards useful alternatives. Second, supervised ranking signals are sharply non-uniform across targets. Some targets are accompanied by dense candidate sets that support score-scale calibration, whereas others provide only two or three labeled candidates per target, in which case continuous regression is statistically unstable and only within-target order is meaningful. Compounding these issues, the reliability of any individual metric is itself target-dependent (Li et al., 2014; Su et al., 2019), so globally fitted weights tend to be brittle and prone to silent leakage across splits when trust is not explicitly modeled.

Existing methods address parts of this problem but not the whole. Classical scoring functions (Koes et al., 2013; Trott & Olson, 2010) score each pose with fixed empirical terms and cannot adapt metric trust to specific targets. Deep affinity predictors (Jiménez et al., 2018; Stepniewska-Dziubinska et al., 2018; Zheng et al., 2019) and graph-based models (Cao et al., 2024; Nguyen et al., 2021; Yang et al., 2023) optimize global regression objectives over large affinity datasets and are tuned for cross-target accuracy rather than within-target ranking. Pose-aware deep rescorers (McNutt et al., 2021; Shen et al., 2022; Zhang et al., 2023) exploit pose information but treat their constituent channels with essentially fixed importance and do not jointly calibrate trust at the suite, metric, and pose levels. Diffusion- and structure-prediction-based affinity models (Corso et al., 2024; Wohlwend et al., 2025) bring stronger structural priors yet typically lack an explicit, leakage-controlled mechanism for adjusting metric trust per target, and they do not separate dense from sparse evaluation regimes. Learning-to-rank formulations have been explored in ligand-based screening (Agarwal et al., 2010;

Zhang et al., 2015), but most operate on aggregated ligand-level features rather than per-pose multi-suite tensors with explicit reliability calibration.

We propose **PoT-Rank** (**Po**se-**T**rust Ranking), a target-local learning-to-rank framework addressing these gaps. Each candidate is represented as a masked pose-by-metric tensor with $P_{\max} = 10$ poses and $Q = 13$ channels partitioned into feature suites (structural confidence, docking affinity, interface geometry, interaction fingerprints, and ligand descriptors). A hierarchical scorer learns softmax weights at the suite and within-suite metric levels with masked pose-attention, and refines them over few rounds. Metric trust is made adaptive without leaking labels through a training-only reliability gate from same-target pairwise concordance, complemented by a target-normalized hard-decoy curriculum that re-weights candidates outranked by weaker binders. Because dense and sparse regimes carry different statistical risks, PoT-Rank splits into two branches: a dense **PoT-Rank-Core** ranker trained with a target-balanced pairwise hinge loss, and a sparse **PoT-Rank-RP-Core** target-balanced antisymmetric pairwise comparator with signed-vote synthesis. All standardization, gates, curriculum weights, cached training-only policy priors, relational memory banks, and inference-time calibration rules are determined without test labels and frozen before evaluation, with leakage audits reported for every stateful component.

Our contributions are fourfold. **(i)** We formulate protein-ligand candidate prioritization as target-local ranking over a pose-by-metric tensor and design a hierarchical scorer that calibrates trust jointly at suite, metric, and pose levels. **(ii)** We introduce a training-only reliability gate from same-target pairwise concordance and a target-normalized hard-decoy curriculum, compatible with a frozen-state protocol. **(iii)** We design dense and sparse branches with manifest-level regime assignment fixed before test evaluation. **(iv)** Under frozen evaluation, PoT-Rank attains the strongest within-target ranking in both regimes, with dense WT-Acc 0.6602, dense Spearman 0.5343, dense Pearson 0.6147, dense $R^2 = 0.4154$, sparse WT-Acc 0.6897, sparse Spearman 0.4313, and sparse Kendall $\tau = 0.4182$, surpassing classical scoring functions, deep affinity predictors, graph-based models, pose-aware rescorers, structure-prediction-based affinity models, while matching or exceeding strong tabular baselines, and remains stable under leakage audits, multi-seed perturbation, and corrupted-channel stress tests.

2 Related Works

Four lines of work bear on target-local ranking, and each leaves one of our two difficulties unaddressed. Classical scoring functions (Trott & Olson, 2010; Koes et al., 2013; Friesner et al., 2004; Gohlke et al., 2000) score each pose with fixed empirical terms, which is fast and interpretable but cannot adapt metric trust per target (Li et al., 2014; Su et al., 2019). Deep affinity predictors and graph-based models (Stepniewska-Dziubinska et al., 2018; Jiménez et al., 2018; Zheng et al., 2019; Nguyen et al., 2021; Yang et al., 2023; Cao et al., 2024) optimize global regression over large affinity datasets, emphasizing cross-target accuracy rather than within-target order, and typically consume a single pose per ligand. Pose-aware rescorers and co-folding models (McNutt et al., 2021; Shen et al., 2022; Zhang et al., 2023; Méndez-Lucio et al., 2021; Corso et al., 2024; 2022; Jumper et al., 2021; Wohlwend et al., 2025) supply rich pose and confidence signals, but combine their channels with fixed or globally learned importance and provide no leakage-controlled mechanism for recalibrating trust per target. Learning-to-rank and pairwise-ensemble approaches (Agarwal et al., 2010; Liu, 2009; Zhang et al., 2015; Boyles et al., 2020; Ericksen et al., 2017) operate on aggregated ligand-level features rather than pose-by-metric tensors, and rarely separate dense from sparse target regimes. PoT-Rank addresses all four gaps jointly.

3 Method

Overview. We propose **PoT-Rank** (**Po**se-**T**rust Ranking), a target-local learning-to-rank framework for protein–ligand candidate prioritization. An overview is shown in Fig. 1. For each protein target, the goal is to rank its candidate ligands using multiple predicted poses of the protein–ligand complex and heterogeneous scoring signals spanning complementary physical scales of binding: co-folding confidence reflecting whether the predicted complex is geometrically trustworthy, docking affinity scoring bound-pose energetics, interface geometry summarizing buried surface and close-contact patterns, interaction fingerprints encoding per-residue contact types (H-bond / hydrophobic / π-

stacking), and ligand descriptors capturing intrinsic chemical properties. Training and evaluation are defined within each target: scores need not be calibrated across unrelated targets, and only within-target order matters.

To keep the paper-level algorithm distinct from benchmark-specific inference calibration, we separate *core* learning components from *retained* variants. **PoT-Rank-Core** denotes the dense pose-aware ranker with training-only reliability gating and a target-normalized hard-decoy curriculum. **PoT-Rank-Policy** adds an optional cached training-only curriculum prior. **PoT-Rank-Policy-Cal** is the retained dense variant that further applies frozen dense memory-control calibration. **PoT-Rank-RP-Core** denotes the sparse target-balanced pairwise comparator, **PoT-Rank-RP + Mem** adds the pairwise relational memory alone, and **PoT-Rank-RP-Full** is the retained sparse variant further adding the sign filter and ambiguity-aware target-local synthesis. The branch assignment between dense and sparse manifests is fixed before test evaluation. Unless otherwise stated, methodological claims in the main text concern **PoT-Rank-Core** and **PoT-Rank-RP-Core**; the construction of all retained benchmark variants is deferred to Appendix A.2. For transparency, the retained variants keep three benchmark-compatibility deviations from the Core equations below, an absolute-value concordance gate, a globally rather than target-normalized curriculum, and a global-mean pairwise hinge, each of which is specified in Appendix A.2.7 and ablated separately.

Notation. Let t index a protein target with candidate set $\mathcal{C}_t$. Each candidate $i \in \mathcal{C}_t$ has supervision y_i, target id t_i, optional scaffold id s_i, and a masked pose tensor $\mathbf{X}_i \in \mathbb{R}^{P_{\max} \times Q}$ with mask $\mathbf{m}_i \in \{0,1\}^{P_{\max}}$, where $P_{\max} = 10$ and $Q = 13$. Labels follow a larger-is-better convention so that $y_i > y_j$ means i should rank above j within the same target. The Q channels are partitioned into $M = 5$ feature suites covering structural confidence, docking affinity, interface geometry, interaction fingerprints, and ligand descriptors. Pose-varying channels remain pose-specific; pose-invariant channels are repeated across valid pose slots. We write the standardized channel value as $x_{i,p,q}$, denote the suite assignment by $\psi(q) \in \{1, \dots, M\}$, and let $\mathcal{Q}_m = \{q : \psi(q) = m\}$. Concretely, the 13 channels are Boltz1 ipTM and mean pLDDT (confidence); Smina and RTMScore affinities (docking); MDAnalysis close contacts and MDTraj BSA (geometry); ProLIF H-bond, hydrophobic, and π-stacking counts (interaction); and RDKit ligand strain, QED, SLogP, and TPSA (ligand property). Their frozen sign conventions are listed in Appendix A.2.4. All standardization statistics are fit on the training split and reused unchanged for validation and test; channel sign conventions are fixed before training and never adapted using evaluation labels.

3.1 Dense PoT-Core

For dense targets, PoT-Rank-Core learns a hierarchical scorer that calibrates trust at the suite, metric, and pose levels. At refinement round r, the suite weights and within-suite metric weights are

$$\alpha_m^{(r)} = \mathrm{softmax}_m\left(u^{(r)}\right), \qquad \beta_{m,q}^{(r)} = \mathrm{softmax}_{q \in \mathcal{Q}_m}\left(v_m^{(r)}\right).$$

A training-only reliability gate $\eta_q \in [0,1]$ (Section 3.2) rescales the within-suite weights and renormalizes them per suite,

$$\tilde{\beta}_{m,q}^{(r)} = \frac{\beta_{m,q}^{(r)} \eta_q}{\sum_{q' \in \mathcal{Q}_m} \beta_{m,q'}^{(r)} \eta_{q'} + \epsilon},$$

falling back to the ungated weights of a suite if numerical clipping or an ablation drives all of its gates to zero. The pose score and the masked pose-attention weights at round r are

$$s_{i,p}^{(r)} = \sum_{m=1}^{M} \alpha_m^{(r)} \sum_{q \in \mathcal{Q}_m} \tilde{\beta}_{m,q}^{(r)} x_{i,p,q}, \qquad \gamma_{i,p}^{(r)} = \frac{m_{i,p} \exp\left(s_{i,p}^{(r)} / \tau^{(r)}\right)}{\sum_{p'} m_{i,p'} \exp\left(s_{i,p'}^{(r)} / \tau^{(r)}\right)}.$$

The final dense score accumulates over R_f refinement rounds,

$$\hat{y}_i = \sum_{r=1}^{R_f} \left[\sum_{p=1}^{P_{\max}} \gamma_{i,p}^{(r)} s_{i,p}^{(r)} + \kappa_i^{(r)} \right],$$

where the optional masked pose-consensus stabilizer $\kappa_i^{(r)} = \zeta^{(r)}\big(1 - \mathrm{Var}_{p \in V_i}[\gamma_{i,p}^{(r)} s_{i,p}^{(r)}] / (\mathrm{Var}_{p \in V_i}[s_{i,p}^{(r)}] + \epsilon)\big)$ rewards candidates whose attention mass concentrates on mutually consistent poses, V_i being the valid-pose set (Appendix A.2.6).

3.2 Training-Only Reliability and Hard-Decoy Curriculum

PoT-Rank-Core estimates metric reliability using only same-target training comparisons, which keeps the calibration target-local. For metric q, the per-sample proxy averages valid poses, $\tilde{x}_{i,q} = |V_i|^{-1} \sum_{p \in V_i} x_{i,p,q}$, with $V_i = \{p : m_{i,p} = 1\}$. For training target t, let $\mathcal{P}_t$ be the set of labeled same-target pairs that survive the gap and metric-tolerance filters. The signed target-level concordance and target-balanced reliability score are

$$a_{t,q} = \frac{1}{|\mathcal{P}_t|} \sum_{(i,j) \in \mathcal{P}_t} \operatorname{sgn}_\delta(\tilde{x}_{i,q} - \tilde{x}_{j,q}) \operatorname{sgn}(y_i - y_j), \qquad \rho_q = \frac{1}{|\mathcal{T}_q|} \sum_{t \in \mathcal{T}_q} a_{t,q},$$

where $\operatorname{sgn}_\delta \in \{-1, 0, +1\}$ uses fixed metric tolerance and $\mathcal{T}_q$ is training targets with at least one eligible pair. The reliability gate is the nonnegative map $\eta_q = \sigma((\rho_q - \tau_{\text{rel}})/T_{\text{rel}})$. Because feature directions are fixed before training, η_q attenuates or preserves metric importance but never flips signs.

After each training cycle, PoT-Rank-Core identifies *hard decoys*, i.e., candidates that are repeatedly outranked by weaker binders within the same target, $\mathcal{D}_i = \{j : t_j = t_i,\ \hat{y}_j > \hat{y}_i,\ y_j < y_i\}$, and updates the next-cycle sample weights through a target-normalized exponential curriculum,

$$w_i^{(c+1)} = \frac{|\mathcal{C}_{t_i}| \exp(\lambda_{\text{hd}} |\mathcal{D}_i|)}{\sum_{j \in \mathcal{C}_{t_i}} \exp(\lambda_{\text{hd}} |\mathcal{D}_j|)}.$$

This normalization keeps the average sample weight equal to 1 inside every target while concentrating later cycles on persistently misranked candidates. The optional cached training-only multiplier of **PoT-Rank-Policy**, $\pi_i \in \{0.90, 0.95, 1.00, 1.05, 1.10\}$, is materialized as a frozen *offline* diagnostic cache (an LLM-generated policy row per training candidate, with provider, model, and schema identifiers logged for reproducibility). It is built from training diagnostics only, contains no validation or test rows, issues no runtime query, and never modifies the dense forward scorer at validation or test time. Its precise rescaling rule and reproducibility metadata are deferred to Appendix A.2.8. The retained dense variant **PoT-Rank-Policy-Cal** adds one further component and one label-free inference rule. The *dense memory* is a frozen training-split store that, for a query candidate, retrieves its nearest training neighbours in the standardized pose-summary space and returns a retrieval confidence $r_i \in [0, 1]$ together with a neighbour-averaged adjustment of the per-metric gates η_q; it never sees validation or test labels. Writing $\hat{y}_i^{\text{mem}}$ and $\hat{y}_i^{\text{base}}$ for the dense scores with and without this adjustment, and $\Delta_t = |\mathcal{C}_t|^{-1} \sum_{i \in \mathcal{C}_t} (\hat{y}_i^{\text{mem}} - \hat{y}_i^{\text{base}})$, the rule keeps the memory branch when $\Delta_t \le 0$ and otherwise reverts to $\hat{y}_i^{\text{base}}$, shifted by a target-level median offset on the low-confidence subset $r_i \le 0.25$ (Appendix A.2.9).

3.3 Dense Objective

PoT-Rank-Core is trained with a target-balanced within-target pairwise hinge loss. Let $\mathcal{P}_t = \{(i, j) : t_i = t_j = t,\ y_i > y_j\}$, let $\mathcal{T}_b$ be the set of targets present in a batch, and let $\omega_{ij} = 1 + \lambda_{\text{gap}} |y_i - y_j| / \max_{(u,v) \in \mathcal{P}_t} |y_u - y_v|$ be an optional target-normalized affinity-gap weight, disabled ($\omega_{ij} = 1$) in both retained recipes unless stated otherwise. We optimize

$$\mathcal{L}_{\text{rank}} = \frac{1}{|\mathcal{T}_b|} \sum_{t \in \mathcal{T}_b} \frac{1}{|\mathcal{P}_t|} \sum_{(i,j) \in \mathcal{P}_t} w_i\, \omega_{ij}\, \max\big(0,\ \mu - (\hat{y}_i - \hat{y}_j)\big).$$

Target balancing prevents large targets from dominating optimization. The objective optionally includes a masked pose-attention entropy regularizer and a same-target scaffold-consistency regularizer, $\mathcal{L}_{\text{PoT-Rank}} = \mathcal{L}_{\text{rank}} + \lambda_{\text{pose}} \mathcal{L}_{\text{pose}} + \lambda_{\text{scaff}} \mathcal{L}_{\text{scaff}}$, whose forms are given in Appendix A.2.6.

3.4 Sparse PoT-RP-Core

When target-local supervision becomes sparse (typically two or three candidates per target), stable dense calibration is infeasible. We therefore use **PoT-Rank-RP-Core**, a sparse-target extension that compresses each tensor into a compact summary vector $\mathbf{z}_i \in \mathbb{R}^D$ and compares candidates pairwise. Concretely, $\mathbf{z}_i$ concatenates, for every pose-varying channel, its $[\text{mean}, \text{std}, \min, \max, \text{span}]$ over valid poses, together with medians of the bounded docking and ligand-strain channels, their

Table 1: Main results on the frozen test splits. WT-Acc denotes within-target pairwise accuracy. The two regimes use disjoint method sets and disjoint metric sets, so they are reported as separate panels; best per column in **bold**, second-best underlined.

(a) Dense regime

Method	Val Acc ↑	Test Acc ↑	Test ρ ↑	Pearson ↑	R^2 ↑
PoT-Rank-Core	0.6060	0.6460	0.1728	0.2785	0.1817
PoT-Rank-Policy	0.6672	0.6566	0.5210	0.6111	0.4081
PoT-Rank-Policy-Cal	0.6701	**0.6602**	**0.5343**	**0.6147**	**0.4154**
XGBoost	**0.6746**	0.5770	0.2737	0.3181	0.1376
CNN	0.6030	0.5558	0.2972	0.2314	0.1183
SVR	0.5716	0.5221	0.1130	0.1663	0.0704

(b) Sparse regime

Method	Val Acc ↑	Test Acc ↑	Test ρ ↑	Kendall τ ↑
PoT-Rank-RP-Core	0.5000	0.5862	0.1667	0.1616
PoT-Rank-RP + Mem	0.5172	0.6552	0.3788	0.3636
PoT-Rank-RP-Full	**0.5259**	**0.6897**	**0.4313**	**0.4182**
XGBoost	0.5000	**0.6897**	0.4202	0.4086
CNN	0.4138	0.6724	0.4010	0.3929
SVR	0.4828	0.4310	−0.0868	−0.1056

cross terms with mean pLDDT, all pose-invariant channels, and the pose-coverage ratio $|V_i|/P_{\max}$ (Appendix A.2.10).

For each same-target unordered pair $\{i, j\}$, we instantiate a single label-independent canonical orientation and define the antisymmetric pair logit

$$h_{ij} = \mathbf{u}_\Delta^\top(\mathbf{z}_i - \mathbf{z}_j), \qquad P(i \succ j) = \sigma(h_{ij}), \qquad P(j \succ i) = 1 - \sigma(h_{ij}).$$

The comparator is trained with a target-balanced binary cross-entropy objective

$$\mathcal{L}_{\mathrm{RP}} = \frac{1}{|\mathcal{T}_b|} \sum_{t \in \mathcal{T}_b} \frac{1}{|\mathcal{P}_t|} \sum_{(i,j) \in \mathcal{P}_t} \omega_{ij}\, \mathrm{BCEWithLogits}(h_{ij}, y_{ij}) + \lambda_1 \|\mathbf{u}_\Delta\|_1 + \lambda_2 \|\mathbf{u}_\Delta\|_2^2.$$

At inference, with $c_{ij} = \mathrm{sign}(h_{ij})$ over the canonical pair table $\mathcal{P}_t^{\mathrm{can}}$, target-local scores are obtained by signed voting, $u_i = \sum_{(u,v) \in \mathcal{P}_t^{\mathrm{can}}} (\mathbb{1}[u{=}i] - \mathbb{1}[v{=}i]) c_{uv}$, normalized by the number n_i of within-target pairs touching i and re-centered within the target, $v_i = u_i/n_i - |\mathcal{C}_t|^{-1} \sum_u u_u/n_u$.

The retained benchmark variant **PoT-Rank-RP-Full** augments this comparator with a frozen training-split pairwise relational memory. Writing $\Delta_{ij} = \mathbf{z}_i - \mathbf{z}_j$ and $\phi_{ij} = [\Delta_{ij}; |\Delta_{ij}|]$, the memory stores for every canonical training pair the residual $r_{ij} = o_{ij} - \tanh(h_{ij}^{\mathrm{base}})$ with $o_{ij} = 2y_{ij} - 1$, keyed by ϕ_{ij}; a query pair retrieves its top-k cosine neighbours and corrects the logit by $h_{ij}^{\mathrm{mem}} = h_{ij}^{\mathrm{base}} + \eta \tanh(\bar{r}_{ij})$, where $\bar{r}_{ij}$ is the similarity-normalized neighbour residual. This single component accounts for the largest sparse ablation delta (-0.10 WT-Acc, Table 3). A deterministic sign filter and an ambiguity-aware synthesis route are applied on top; the route fires only when $|\mathcal{C}_t| = 3$, at least one pair flips from a negative base logit to a positive memory logit, no pair flips in the reverse direction, and the mean ambiguous-logit magnitude satisfies $m_t \geq 0.02$, with the threshold selected on validation and frozen before test evaluation (Appendices A.2.11–A.2.13).

3.5 Frozen-State Protocol

PoT-Rank is trained and evaluated under a frozen-state protocol; training and inference are summarized in Algorithms 1–2 (Appendix A.2.2). Feature standardization, reliability gates, curriculum weights, cached policy priors, relational memory banks, and inference-time calibration rules are constructed without test labels and frozen before final test evaluation. Test labels are used only for final metrics. Per-component construction rules and counterfactual leakage audits, in which each component is intentionally rebuilt with information unavailable under the protocol, are given in Appendices A.2.14 and A.7.

4 Experiments and Results

4.1 Setup

Datasets and regimes. We evaluate PoT-Rank on two regimes: a *dense-target* regime with few targets but large candidate sets, and a *sparse-target* regime with many targets but very small candidate sets. To prevent train/test leakage from homologous proteins, all train / val / test splits are constructed on the *protein side* using MMseqs2 (Steinegger & Söding, 2017) clustering at sequence-identity threshold 0.4 (i.e., proteins sharing $\geq 40\%$ identity are forced into the same split), so no

Table 2: Comparison with domain-specific scorers and learned affinity baselines on the frozen dense test split. The four rightmost columns use target-median relative actives and unweighted target-macro averaging. Commercially gated engines (Glide SP/XP) are omitted because no license was available under this protocol.

Method	**Ranking & Correlation**				**Target-relative Diagnostics**			
	Test Acc ↑	Test ρ ↑	Pearson ↑	R^2 ↑	Rel. EF1% ↑	Rel. EF5% ↑	$\text{BEDROC}_{80.5}$ ↑	ROC-AUC ↑
Classical scoring functions								
AutoDock Vina	0.5150	−0.0598	0.0488	0.0319	0.00	0.68	0.0215	0.3593
Smina	0.5133	−0.0631	0.0546	0.0333	0.00	0.68	0.0215	0.3582
Deep affinity predictors								
Pafnucy	0.5009	−0.0165	−0.0224	0.0134	0.00	0.68	0.0222	0.5070
OnionNet	0.5929	0.1218	0.1101	0.0752	**2.12**	1.43	0.9779	0.5718
K_{DEEP}	0.5451	0.0827	−0.0375	0.0115	0.00	0.33	0.0145	0.5648
Graph-based models								
GraphDTA	0.5106	−0.0775	−0.2240	0.1157	0.00	0.35	0.0079	0.5094
IGN	0.5274	0.1213	0.0669	0.0170	1.45	**1.45**	0.6671	0.5892
GIGN	0.5274	0.0668	0.0645	0.0451	1.45	1.43	0.6738	0.5277
EquiScore	0.5133	0.0037	−0.0365	0.0064	0.67	0.68	0.3254	0.4452
Pose-aware deep models								
GNINA	0.5080	−0.0108	−0.1672	0.0604	0.00	0.33	0.0146	0.5321
RTMScore	0.4761	−0.2298	−0.1167	0.0766	1.37	1.02	0.6594	0.3197
KarmaDock	0.5274	0.0551	0.0760	0.0510	1.37	0.68	0.6452	0.4420
Diffusion / structure-prediction-based								
DiffDock-affinity	0.5080	0.0842	0.0095	0.0183	1.45	**1.45**	0.6666	0.5458
AlphaFold-Latent-Aff	0.4903	−0.0850	−0.2507	0.1147	0.00	0.68	0.0217	0.4960
PoT-Rank-Policy-Cal (ours)	**0.6602**	**0.5343**	**0.6147**	**0.4154**	**2.12**	1.43	**0.9785**	**0.7474**

test target shares a $\geq 40\%$-identity cluster with any training target. Regime assignment is fixed at the manifest level before test evaluation by a frozen rule on *training-split* target-size statistics only; manifests matching neither rule require an explicit pre-declared branch, and auto-routing is disabled (Appendix A.2.3). After the protocol-required reliability filter, the dense regime retains 8 / 2 / 3 targets across train / val / test with average candidate-set size $|\mathcal{C}_t| \approx 22$, and the sparse regime retains 659 / 73 / 79 targets with $|\mathcal{C}_t| \approx 1.7$. The two regimes differ by more than an order of magnitude in candidate-set size, which is what motivates the algorithmic split of Section 3; per-split ligand and pose counts are in Appendix A.3.

Pose generation. Boltz-1 (Wohlwend et al., 2025) predicts each complex and defines the docking box; AutoDock Vina (Trott & Olson, 2010) generates up to ten poses per candidate, which Smina (Koes et al., 2013) and RTMScore then *rescore* rather than re-dock, so all thirteen channels describe the same pose population. Full tool settings are in Appendix A.2.5.

Baselines. We compare PoT-Rank against three families.

(i) Tabular baselines: XGBoost, CNN, and SVR, reproduced under the same partitions for both regimes.

(ii) Structure-based scorers, dense regime only: five families covering classical scoring functions, deep affinity predictors, graph-based models, pose-aware deep rescorers, and diffusion / structure-prediction-based predictors; the fourteen individual methods are listed in Table 2. Where a reusable reference pipeline is unavailable, the comparison uses a protocol-matched reimplementation under the same input/output contract, and commercially gated methods such as Glide SP/XP are reported only when runtime and license are available.

(iii) PoT-Rank family: the dense variants of Section 3 (PoT-Rank-Core, PoT-Rank-Policy, PoT-Rank-Policy-Cal; retained: PoT-Rank-Policy-Cal) and the sparse variants (PoT-Rank-RP-Core, PoT-Rank-RP + Mem, PoT-Rank-RP-Full; retained: PoT-Rank-RP-Full). The full glossary is in Appendix A.4.

PoT-Rank-RP+Mem**Evaluation metrics.** The primary metric is within-target pairwise accuracy (**WT-Acc**). For the dense regime we additionally report target-mean Spearman (ρ), Pearson (r), and R^2; for the sparse regime we report target-mean Spearman and Kendall τ in the main table, with overall Pair Acc reported alongside the sparse ablations (Table 3). For the dense comparison, we further report relative EF1%, relative EF5%, $\text{BEDROC}_{80.5}$, and ROC-AUC. Within each target, ligands whose measured affinity exceeds the target median are treated as relative actives; each metric

Table 3: Component ablations on the dense (top) and sparse (bottom) retained variants. Δ Acc reports test WT-Acc change relative to the retained variant. Component IDs are explained in Appendix A.4. In the sparse block, the columns headed "Pearson / τ" and "R^2 / PairAcc" report Kendall τ and overall Pair Acc.

Variant	Val Acc ↑	Test Acc ↑	Test ρ ↑	Pearson / τ ↑	R^2 / PairAcc ↑	Δ Acc
Dense regime — reference: PoT-Rank-Policy-Cal						
PoT-Rank-Policy-Cal	**0.6701**	**0.6602**	**0.5343**	**0.6147**	**0.4154**	0.0000
− D1	0.6672	0.6566	0.5210	0.6111	0.4081	−0.0035
− D2	0.6687	0.6549	0.5273	0.6081	0.4066	−0.0053
− D3	0.6657	0.6496	0.5151	0.6087	0.4065	−0.0106
− D4	0.6672	0.6566	0.5210	0.6111	0.4081	−0.0035
− D5	0.6060	0.6460	0.1728	0.2785	0.1817	−0.0142
Sparse regime — reference: PoT-Rank-RP-Full						
PoT-Rank-RP-Full	**0.5259**	**0.6897**	**0.4313**	**0.4182**	**0.5582**	0.0000
− S1	0.5259	0.6810	0.4202	0.4035	0.5586	−0.0086
− S2	0.5172	0.6724	0.4050	0.3884	0.5566	−0.0172
− S3	0.5172	0.6552	0.3788	0.3636	0.5548	−0.0345
− S4	0.5000	0.5862	0.1667	0.1616	0.5413	−0.1035
− S5	0.5000	0.6379	0.3415	0.3344	0.5513	−0.0518
− S6	0.5000	0.6552	0.3677	0.3591	0.5531	−0.0345
− S7	0.5000	0.6207	0.3374	0.3187	0.5500	−0.0690
− S8	0.4914	0.6034	0.3182	0.3030	0.5412	−0.0863

is computed within target and then averaged equally across targets. We use these quantities as descriptive target-relative ranking diagnostics. All reported metrics are higher-is-better.

Implementation details. All PoT-Rank variants use $P_{\max}$=10 poses, Q=13 channels, and M=5 feature suites. The dense recipe uses R_f=5 refinement rounds, hard-decoy curriculum strength λ_{hd}=1.0, and dense confidence threshold 0.25; the sparse recipe uses memory neighborhood k=5, residual scale η=0.2, and ambiguity-router thresholds $m_t \geq 0.02$ with size-3 flip weight α=3.0. Standardization statistics are fit on the training split and reused unchanged for validation and test. All experiments run on a single NVIDIA RTX 3060 Laptop GPU with PyTorch 3.11.9.

Frozen-state protocol. All stateful PoT-Rank components are frozen as specified in Section 3.5. Counterfactual leakage audits are reported in Appendix A.7: only two components are leakage-sensitive, the dense reliability gate (rebuilding it with leaked labels would inflate test WT-Acc/ρ by $+0.005$/$+0.128$) and the sparse relational memory (leaked pair tables would shift WT-Acc/ρ by -0.043/-0.102); all other stateful components are audit-invariant. We flag one limitation of this protocol: sparse validation accuracy sits near chance for every method (Table 1b), so the sparse ambiguity-router threshold is best understood as a coarse, pre-registered choice rather than a validated one; Appendix A.2.13 reports that the retained band $0.02 \leq m_t < 0.03$ and the original $m_t \geq 0.03$ route differ by -0.0086 test WT-Acc (S1), which bounds the sensitivity to this choice.

4.2 Main Results

Table 1 reports the headline comparison. The retained dense variant PoT-Rank-Policy-Cal achieves the best score on every dense *test* metric, improving over PoT-Rank-Core by $+0.014$ WT-Acc and by $+0.36$, $+0.34$, and $+0.23$ in test ρ, Pearson, and R^2, respectively. This indicates that the cached training-only prior and frozen dense memory-control calibration contribute beyond the dense core. On the sparse split, the gain stack PoT-Rank-RP-Core → PoT-Rank-RP + Mem → PoT-Rank-RP-Full produces a $+0.10$ test WT-Acc improvement, with relational memory contributing the largest single delta. Against the strongest tabular baseline, PoT-Rank-Policy-Cal exceeds dense XGBoost by $+0.083$ WT-Acc and $+0.261$ Spearman, while PoT-Rank-RP-Full matches sparse XGBoost on WT-Acc and slightly exceeds it on target-local ρ and τ. XGBoost does attain the highest dense *validation* accuracy (0.6746) while ranking below PoT-Rank on every dense test metric; with two validation targets the validation surface cannot separate methods, which is why model selection

is fixed by the frozen protocol rather than by validation score. Multi-seed verification, target-size stratification, and extended hyper-parameter sweeps are reported in Appendix A.6, Appendix A.10, and Appendix A.9, respectively.

4.3 Comparison with Domain-Specific and Learned Affinity Methods

Table 2 compares PoT-Rank-Policy-Cal with classical scoring functions, deep affinity predictors, graph-based models, pose-aware deep models, and diffusion / structure-prediction-based predictors on the frozen dense test split. PoT-Rank-Policy-Cal is strongest across every reported ranking and correlation metric: the gap to the best non-PoT-Rank learned baseline is $+0.067$ WT-Acc and $+0.41$ Spearman over OnionNet. Under the target-relative evaluation, PoT-Rank-Policy-Cal ties OnionNet for the highest relative EF1% (2.12), attains the highest $\text{BEDROC}_{80.5}$ (0.9785) and ROC-AUC (0.7474), and remains close to the highest relative EF5% (1.43 versus 1.45). These descriptive top-ranking diagnostics are consistent with the primary within-target ranking results.

4.4 Ablation Studies

Table 3 reports frozen-protocol ablations; dense IDs D1–D5 and sparse IDs S1–S8 are defined in Appendix A.4. In the dense regime, the eval-time low-confidence router gives the largest single-component drop (-0.011 WT-Acc), whereas jointly removing the cached policy prior and calibration layer (D5) causes a larger aggregate loss (-0.014 WT-Acc, -0.36 test ρ), indicating that these components mainly correct score *scale*. In the sparse regime, the relational memory bank is most important (-0.10 WT-Acc), followed by the target-balanced objective (-0.086) and signed-vote synthesis (-0.069). Confidence-cross features and signed-vote synthesis interact, jointly approaching the no-memory floor when removed. Feature-suite leave-one-out results (Appendix A.8) show that no single cue is sufficient: removing Boltz-1 confidence and RDKit descriptors costs -0.120 and -0.117 WT-Acc, respectively, while ProLIF receives the largest final-round weight (α_k=0.886) but the smallest leave-one-out drop, so learned weights should not be interpreted causally. Stratification of the 565 dense test pairs (Appendix A.5) shows that gains over OnionNet concentrate in close-affinity pairs ($+0.032$ WT-Acc at $|\Delta pK_d| \leq 0.5$), activity-cliff analogs ($+0.089$ at Tanimoto ≥ 0.6), and interaction-rich poses ($+0.090$ at ≥ 10 unique residue contacts).

4.5 Stability and Cost

Multi-seed runs (n=5) confirm stability: sparse PoT-Rank-RP-Full is deterministic under the retained recipe (σ=0), while dense PoT-Rank-Policy-Cal averages 0.6336 ± 0.0222 WT-Acc and 0.4541 ± 0.0776 ρ. The headline dense value of 0.6602 is the single retained seed fixed before test evaluation and sits above this mean, so the multi-seed average is the conservative reading; under either reading the gap over repeated XGBoost is $+0.058$ WT-Acc and $+0.21$ ρ. Frozen-inference robustness holds under σ=0.5 Gaussian noise and 30% random masking, with WT-Acc within ± 0.015 of the clean reference. The whole system is light: on one RTX 3060 Laptop, training takes 26.85 s (dense) and 38.69 s (sparse), and a typical dense target is served in ~ 1.1 s at $|\mathcal{C}_t|$=50; the cached prior adds no runtime cost, being a frozen offline cache. Robustness panels, leave-one-out, hyper-parameter sweeps, leakage audits and latency curves are in Appendices A.6–A.12.

5 Conclusion and Outlook

We presented PoT-Rank, a target-local learning-to-rank framework that encodes candidates as masked pose-by-metric tensors and learns hierarchical trust weights over suites, metrics, and poses via a reliability gate and hard-decoy curriculum. Its dense PoT-Rank-Core ranker and sparse PoT-Rank-RP-Core comparator achieve the best within-target ranking on both splits (dense WT-Acc 0.6602; sparse WT-Acc 0.6897), outperforming classical scoring, deep affinity, graph-based, pose-aware, and structure-prediction baselines. Gains concentrate in close-affinity, activity-cliff, and interaction-rich cases where classical scoring often fails (Appendix A.5). Future work includes adding MD-derived dynamics and free-energy surrogates, replacing the dense/sparse split with a continuous formulation over $|\mathcal{C}_t|$, enabling conformal selective abstention, and learning calibration rules end-to-end under the frozen-state protocol. Limitations, including dense evaluation scale, are discussed in Appendix A.13.

AI USE STATEMENT

We used AI assistance to polish wording and improve the organization and readability of the manuscript. The scientific claims, experimental design, data processing choices, reported results, and final text were reviewed and approved by the authors.

ETHICS STATEMENT

This work uses existing molecular and protein–ligand data and does not involve human subjects or personally identifiable information. The intended use is computational prioritization for structure-based virtual screening. As with other virtual-screening methods, outputs should be interpreted as decision-support signals rather than experimental validation.

REPRODUCIBILITY STATEMENT

We provide the code link in the abstract and describe the frozen-state protocol, data splits, feature construction, training objectives, hyperparameters, ablations, and computational cost in the main text and appendix.

REFERENCES

Shivani Agarwal, Deepak Dugar, and Shiladitya Sengupta. Ranking chemical structures for drug discovery: a new machine learning approach. *Journal of chemical information and modeling*, 50 (5):716–731, 2010.

Anastasios N Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023.

Brian J Bender, Stefan Gahbauer, Andreas Luttens, Jiankun Lyu, Chase M Webb, Reed M Stein, Elissa A Fink, Trent E Balius, Jens Carlsson, John J Irwin, et al. A practical guide to large-scale docking. *Nature protocols*, 16(10):4799–4832, 2021.

Cédric Bouysset and Sébastien Fiorucci. Prolif: a library to encode molecular interactions as fingerprints. *Journal of cheminformatics*, 13(1):72, 2021.

Fergus Boyles, Charlotte M Deane, and Garrett M Morris. Learning from the ligand: using ligand-based features to improve binding affinity prediction. *Bioinformatics*, 36(3):758–764, 2020.

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*, pp. 89–96, 2005.

Duanhua Cao, Geng Chen, Jiaxin Jiang, Jie Yu, Runze Zhang, Mingan Chen, Wei Zhang, Lifan Chen, Feisheng Zhong, Yingying Zhang, et al. Generic protein–ligand interaction scoring by integrating physical prior knowledge and data augmentation modelling. *Nature Machine Intelligence*, 6(6):688–700, 2024.

Paul S Charifson, Joseph J Corkery, Mark A Murcko, and W Patrick Walters. Consensus scoring: A method for obtaining improved hit rates from docking databases of three-dimensional structures into proteins. *Journal of medicinal chemistry*, 42(25):5100–5109, 1999.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. In *arXiv preprint arXiv:2210.01776*, 2022.

Gabriele Corso, Arthur Deng, Nicholas Polizzi, Regina Barzilay, and Tommi Jaakkola. Deep confident steps to new pockets: Strategies for docking generalization. In *International Conference on Learning Representations*, volume 2024, pp. 50377–50397, 2024.

Spencer S Ericksen, Haozhen Wu, Huikun Zhang, Lauren A Michael, Michael A Newton, F Michael Hoffmann, and Scott A Wildman. Machine learning consensus scoring improves performance across targets in structure-based virtual screening. *Journal of chemical information and modeling*, 57(7):1579, 2017.

Richard Evans, Michael O'neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Tim Green, Augustin Žídek, Russ Bates, Sam Blackwell, Jason Yim, et al. Protein complex prediction with alphafold-multimer. *biorxiv*, pp. 2021–10, 2021.

Richard A Friesner, Jay L Banks, Robert B Murphy, Thomas A Halgren, Jasna J Klicic, Daniel T Mainz, Matthew P Repasky, Eric H Knoll, Mee Shelley, Jason K Perry, et al. Glide: a new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy. *Journal of medicinal chemistry*, 47(7):1739–1749, 2004.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Holger Gohlke, Manfred Hendlich, and Gerhard Klebe. Knowledge-based scoring function to predict protein-ligand interactions. *Journal of molecular biology*, 295(2):337–356, 2000.

Lior Hirschfeld, Kyle Swanson, Kevin Yang, Regina Barzilay, and Connor W Coley. Uncertainty quantification using neural networks for molecular property prediction. *Journal of Chemical Information and Modeling*, 60(8):3770–3780, 2020.

Douglas R Houston and Malcolm D Walkinshaw. Consensus docking: improving the reliability of docking in a virtual screening context. *Journal of chemical information and modeling*, 53(2): 384–390, 2013.

Sheng-You Huang and Xiaoqin Zou. An iterative knowledge-based scoring function to predict protein–ligand interactions: I. derivation of interaction potentials. *Journal of computational chemistry*, 27(15):1866–1875, 2006.

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

José Jiménez, Miha Skalic, Gerard Martinez-Rosell, and Gianni De Fabritiis. K deep: protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. *Journal of chemical information and modeling*, 58(2):287–296, 2018.

Gareth Jones, Peter Willett, Robert C Glen, Andrew R Leach, and Robin Taylor. Development and validation of a genetic algorithm for flexible docking. *Journal of molecular biology*, 267(3): 727–748, 1997.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*, volume 30, 2017.

David Ryan Koes, Matthew P Baumgartner, and Carlos J Camacho. Lessons learned in empirical scoring with smina from the csar 2011 benchmarking exercise. *Journal of chemical information and modeling*, 53(8):1893, 2013.

Rohith Krishna, Jue Wang, Woody Ahern, Pascal Sturmfels, Preetham Venkatesh, Indrek Kalvet, Gyu Rie Lee, Felix S Morey-Burrows, Ivan Anishchenko, Ian R Humphreys, et al. Generalized biomolecular modeling and design with rosettafold all-atom. *Science*, 384(6693):eadl2528, 2024.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, volume 30, 2017.

Shuangli Li, Jingbo Zhou, Tong Xu, Liang Huang, Fan Wang, Haoyi Xiong, Weili Huang, Dejing Dou, and Hui Xiong. Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 975–985, 2021.

Yan Li, Li Han, Zhihai Liu, and Renxiao Wang. Comparative assessment of scoring functions on an updated benchmark: 2. evaluation methods and general results. *Journal of chemical information and modeling*, 54(6):1717–1736, 2014.

Tie-Yan Liu. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

Zhihai Liu, Minyi Su, Li Han, Jie Liu, Qifan Yang, Yan Li, and Renxiao Wang. Forging the basis for developing protein–ligand interaction scoring functions. *Accounts of chemical research*, 50 (2):302–309, 2017.

Robert T McGibbon, Kyle A Beauchamp, Matthew P Harrigan, Christoph Klein, Jason M Swails, Carlos X Hernández, Christian R Schwantes, Lee-Ping Wang, Thomas J Lane, and Vijay S Pande. Mdtraj: a modern open library for the analysis of molecular dynamics trajectories. *Biophysical journal*, 109(8):1528–1532, 2015.

Andrew T McNutt, Paul Francoeur, Rishal Aggarwal, Tomohide Masuda, Rocco Meli, Matthew Ragoza, Jocelyn Sunseri, and David Ryan Koes. Gnina 1.0: molecular docking with deep learning. *Journal of cheminformatics*, 13(1):43, 2021.

Oscar Méndez-Lucio, Mazen Ahmad, Ehecatl Antonio del Rio-Chanona, and Jörg Kurt Wegner. A geometric deep learning approach to predict binding conformations of bioactive molecules. *Nature Machine Intelligence*, 3(12):1033–1039, 2021.

Naveen Michaud-Agrawal, Elizabeth J Denning, Thomas B Woolf, and Oliver Beckstein. Mdanalysis: a toolkit for the analysis of molecular dynamics simulations. *Journal of computational chemistry*, 32(10):2319–2327, 2011.

Seokhyun Moon, Wonho Zhung, Soojung Yang, Jaechang Lim, and Woo Youn Kim. Pignet: a physics-informed deep learning model toward generalized drug–target interaction predictions. *Chemical Science*, 13(13):3661–3673, 2022.

Gerd Neudert and Gerhard Klebe. Dsx: a knowledge-based scoring function for the assessment of protein–ligand complexes. *Journal of chemical information and modeling*, 51(10):2731–2745, 2011.

Thin Nguyen, Hang Le, Thomas P Quinn, Tri Nguyen, Thuc Duy Le, and Svetha Venkatesh. Graphdta: predicting drug–target binding affinity with graph neural networks. *Bioinformatics*, 37(8):1140–1147, 2021.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In *Advances in neural information processing systems*, volume 31, 2018.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *arXiv preprint arXiv:1701.06538*, 2017.

Chao Shen, Xujun Zhang, Yafeng Deng, Junbo Gao, Dong Wang, Lei Xu, Peichen Pan, Tingjun Hou, and Yu Kang. Boosting protein–ligand binding pose prediction and virtual screening based on residue–atom distance likelihood potential and graph transformer. *Journal of Medicinal Chemistry*, 65(15):10691–10706, 2022.

Brian K Shoichet. Virtual screening of chemical libraries. *Nature*, 432(7019):862–865, 2004.

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

Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

Marta M Stepniewska-Dziubinska, Piotr Zielenkiewicz, and Pawel Siedlecki. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. *Bioinformatics*, 34(21):3666–3674, 2018.

Minyi Su, Qifan Yang, Yu Du, Guoqin Feng, Zhihai Liu, Yan Li, and Renxiao Wang. Comparative assessment of scoring functions: the casf-2016 update. *Journal of chemical information and modeling*, 59(2):895–913, 2019.

Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

Mikhail Volkov, Joseph-André Turk, Nicolas Drizard, Nicolas Martin, Brice Hoffmann, Yann Gaston-Mathé, and Didier Rognan. On the frustration to predict binding affinities from protein–ligand structures with deep neural networks. *Journal of medicinal chemistry*, 65(11):7946–7958, 2022.

Izhar Wallach, Michael Dzamba, and Abraham Heifets. Atomnet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. In *arXiv preprint arXiv:1510.02855*, 2015.

Jeremy Wohlwend, Gabriele Corso, Saro Passaro, Noah Getz, Mateo Reveiz, Ken Leidal, Wojtek Swiderski, Liam Atkinson, Tally Portnoi, Itamar Chinn, et al. Boltz-1 democratizing biomolecular interaction modeling. *BioRxiv*, pp. 2024–11, 2025.

Maciej Wójcikowski, Pedro J Ballester, and Pawel Siedlecki. Performance of machine-learning scoring functions in structure-based virtual screening. *Scientific Reports*, 7(1):46710, 2017.

Jincai Yang, Cheng Shen, and Niu Huang. Predicting or pretending: artificial intelligence for protein-ligand interactions lack of sufficiently large and unbiased datasets. *Frontiers in pharmacology*, 11:69, 2020.

Ziduo Yang, Weihe Zhong, Qiujie Lv, Tiejun Dong, and Calvin Yu-Chian Chen. Geometric interaction graph neural network for predicting protein–ligand binding affinities from 3d structures (gign). *The journal of physical chemistry letters*, 14(8):2020–2033, 2023.

Wei Zhang, Lijuan Ji, Yanan Chen, Kailin Tang, Haiping Wang, Ruixin Zhu, Wei Jia, Zhiwei Cao, and Qi Liu. When drug discovery meets web search: learning to rank for ligand-based virtual screening. *Journal of cheminformatics*, 7(1):5, 2015.

Xujun Zhang, Odin Zhang, Chao Shen, Wanglin Qu, Shicheng Chen, Hanqun Cao, Yu Kang, Zhe Wang, Ercheng Wang, Jintu Zhang, et al. Efficient and accurate large library ligand docking with karmadock. *Nature Computational Science*, 3(9):789–804, 2023.

Liangzhen Zheng, Jingrong Fan, and Yuguang Mu. Onionnet: a multiple-layer intermolecular-contact-based convolutional neural network for protein–ligand binding affinity prediction. *ACS omega*, 4(14):15956, 2019.

A Appendix

A.1 Related Work

Classical scoring functions and docking engines. Empirical, force-field, and knowledge-based scoring functions are the long-standing backbone of structure-based virtual screening. AutoDock Vina (Trott & Olson, 2010) and its derivative Smina (Koes et al., 2013) use empirical pose energetics for fast docking and rescoring; commercial engines such as Glide (Friesner et al., 2004) and GOLD (Jones et al., 1997) integrate sampling with empirical scoring; and knowledge-based potentials including DrugScore (Gohlke et al., 2000), ITScore (Huang & Zou, 2006), and DSX (Neudert

& Klebe, 2011) fit pairwise statistical preferences from known complexes. CASF benchmarks (Li et al., 2014; Su et al., 2019) have shown that no single classical function dominates across all targets, motivating consensus and rescoring strategies. These methods are interpretable and efficient but apply uniform term weights across targets and pose populations, which limits their within-target ranking accuracy.

Deep learning for binding-affinity regression. A second line of work replaces empirical terms with learned representations. CNN-based predictors operate on voxelized pockets, including Pafnucy (Stepniewska-Dziubinska et al., 2018), K_{DEEP} (Jiménez et al., 2018), AtomNet (Wallach et al., 2015), and OnionNet (Zheng et al., 2019). Graph- and equivariant-network-based models such as GraphDTA (Nguyen et al., 2021), IGN (Yang et al., 2023), SIGN (Li et al., 2021), GIGN (Yang et al., 2023), PIGNet (Moon et al., 2022), and EquiScore (Cao et al., 2024) encode protein-ligand graphs with topology-aware message passing. Most of these models are trained as global regressors over PDBbind-scale datasets (Liu et al., 2017) and are evaluated by cross-target metrics. As several analyses have noted (Volkov et al., 2022; Yang et al., 2020), strong cross-target R^2 does not translate to reliable within-target ordering, especially under low-data regimes, and these models typically use a single pose per ligand.

Pose-aware rescoring and structure-aware affinity. Pose-aware deep rescorers exploit structural information at the candidate level. GNINA (McNutt et al., 2021) couples a CNN scorer with conventional docking; RTMScore (Shen et al., 2022) uses residue-atom likelihoods with a graph transformer; KarmaDock (Zhang et al., 2023) unifies docking and scoring; DeepDock (Méndez-Lucio et al., 2021) learns mesh-based pose potentials. More recently, diffusion-based DiffDock (Corso et al., 2024; 2022) and co-folding models such as AlphaFold (Jumper et al., 2021), AlphaFold-Multimer (Evans et al., 2021), RoseTTAFold All-Atom (Krishna et al., 2024), and Boltz-1 (Wohlwend et al., 2025) predict structures together with confidence signals usable as ranking features. While these systems supply rich pose-level evidence, the constituent channels are usually fused with fixed weights or with weights learned globally, without explicit, leakage-controlled mechanisms for adjusting trust on a per-target basis.

Learning-to-rank, consensus, and ensemble scoring. Ranking objectives have a long history in information retrieval (Burges et al., 2005; Liu, 2009) and have been adapted to ligand-based screening (Agarwal et al., 2010; Zhang et al., 2015) and to docking via pairwise classifiers and gradient-boosted ensembles (Boyles et al., 2020; Ericksen et al., 2017; Wójcikowski et al., 2017). Consensus scoring (Charifson et al., 1999; Houston & Walkinshaw, 2013) averages or votes across heterogeneous functions, and feature-stacked tabular models such as XGBoost (Chen & Guestrin, 2016), LightGBM (Ke et al., 2017), and CatBoost (Prokhorenkova et al., 2018) are common strong baselines. These approaches are typically built on ligand-level aggregated features and either treat each candidate independently or learn a single global combination, leaving pose-level structure, target-conditional reliability, and dense versus sparse regimes unmodeled.

Reliability calibration and trust-aware fusion. A complementary thread aims to weight evidence by its reliability. Mixture-of-experts (Jacobs et al., 1991; Shazeer et al., 2017) and attention-based fusion learn input-dependent weights, while uncertainty-aware ensembling (Gal & Ghahramani, 2016; Lakshminarayanan et al., 2017) and conformal prediction (Angelopoulos & Bates, 2023) attach confidence to predictions. In molecular property prediction, evidential and reliability-weighted approaches have shown gains under distribution shift (Hirschfeld et al., 2020; Soleimany et al., 2021). PoT-Rank aligns with this line by introducing a training-only reliability gate based on same-target pairwise concordance and a target-normalized hard-decoy curriculum, but differs in operating directly on a pose-by-metric tensor with explicit suite-, metric-, and pose-level calibration, and in committing to a frozen-state protocol with counterfactual leakage audits across both dense and sparse evaluation regimes.

A.2 Method Details and Retained Benchmark Variants

This appendix collects implementation-level details of PoT-Rank that are kept out of the main text for clarity. Section A.2.1 provides the variant map. Sections A.2.3 and A.2.4 document the manifest-level regime assignment and the feature inventory. The remaining subsections specify the auxiliary

terms of the dense ranker (Section A.2.6), the retained dense benchmark recipe (Section A.2.7), the cached policy prior together with its reproducibility metadata (Section A.2.8), the dense memory-control calibration (Section A.2.9), the sparse summary vector and canonical pair construction (Section A.2.10), the sparse relational memory (Section A.2.11), the sparse target-local synthesis (Section A.2.12), the validation-selected sparse sign filter and ambiguity router (Section A.2.13), and the expanded leakage protocol (Section A.2.14).

A.2.1 Variant Map

PoT-Rank-Core is the dense core algorithm and consists of the pose-aware hierarchical scorer, the signed reliability gate, the target-normalized hard-decoy curriculum, and the target-balanced hinge loss. PoT-Rank-Policy is the dense intermediate variant that adds the cached training-only curriculum prior of Section A.2.8. PoT-Rank-Policy-Cal is the retained dense benchmark variant that combines PoT-Rank-Policy with the conservative nonnegative benchmark gate, the frozen dense memory-control calibration, and the retained dense compatibility training recipe of Section A.2.7. PoT-Rank-RP-Core is the sparse core algorithm, consisting of the target-balanced antisymmetric comparator with default signed-vote synthesis. PoT-Rank-RP + Mem augments PoT-Rank-RP-Core with the pairwise relational memory of Section A.2.11, with the sign filter and ambiguity router disabled. PoT-Rank-RP-Full is the retained sparse benchmark variant that further adds the deterministic sign filter and the ambiguity-aware target-local synthesis of Section A.2.13.

A.2.2 Training and Inference Pseudocode

Algorithms 1 and 2 summarize the PoT-Rank training and per-target inference procedures for both the dense and sparse branches under the frozen-state protocol of Section 3.5.

Algorithm 1 PoT-Rank training (Core variants).

Require: Training targets $\{\mathcal{C}_t\}$, labels y, pose tensors $\mathbf{X}$, masks $\mathbf{m}$
Fit feature standardization on the training split.
Select dense or sparse branch from the fixed regime assignment (Appendix A.2.3).
if dense branch **then**
  for cycle $c = 1, \ldots, C$ **do**
    Optimize PoT-Rank-Core with the target-balanced hinge loss $\mathcal{L}_{\mathrm{rank}}$.
    Re-estimate training-only reliability gates $\{\eta_q\}$ from same-target pairwise concordance.
    Update the target-normalized hard-decoy curriculum $\{w_i^{(c+1)}\}$.
    (PoT-Rank-Policy/Policy-Cal) optionally rescale by the cached training-only prior $\{\pi_i\}$.
  end for
  (PoT-Rank-Policy-Cal) freeze dense memory-control calibration constants (Appendix A.2.9).
else
  Build the target-balanced canonical pair table.
  Optimize PoT-Rank-RP-Core with $\mathcal{L}_{\mathrm{RP}}$.
  (PoT-Rank-RP-Full) build the frozen training-split relational memory and freeze ambiguity-router thresholds (Appendices A.2.11–A.2.13).
end if
Freeze all parameters, gates, memory banks, and inference rules.
return Frozen PoT-Rank model.

A.2.3 Manifest-Level Regime Assignment

Dense and sparse regimes are assigned at the manifest level by a frozen protocol that combines training-split target-size statistics with manifest-provided regime hints; legacy dataset nicknames such as "small" and "large" are bundle identifiers only. When an explicit regime-family hint is available, it takes precedence. Otherwise, the heuristic fallback used by the retained code path classifies a manifest as `sparse_target` when the training-split statistics satisfy

$$|\mathcal{T}_{\mathrm{train}}| \geq 100, \quad \max_t |\mathcal{C}_t| \leq 4, \quad \mathrm{mean}_t |\mathcal{C}_t| \leq 3.5, \quad \mathrm{median}_t |\mathcal{C}_t| \leq 3,$$

Algorithm 2 PoT-Rank inference for one target.

Require: Candidate set $\mathcal{C}_t$ and frozen PoT-Rank state
1: Build pose tensors and apply frozen training-split standardization.
2: Select the pre-specified dense or sparse branch.
3: **if** dense branch **then**
4: Compute PoT-Rank-Core scores $\{\hat{y}_i\}$.
5: **(PoT-Rank-Policy-Cal)** apply frozen dense memory-control calibration to obtain $\{\tilde{y}_i\}$.
6: **else**
7: Compute PoT-Rank-RP-Core pair logits $\{h_{ij}\}$.
8: **(PoT-Rank-RP-Full)** apply the frozen relational memory, sign filter, and ambiguity-aware synthesis.
9: Synthesize target-local scores by signed voting (or weighted signed voting on routed targets).
10: **end if**
11: **Return** the within-target ranking induced by the final scores.

Table 4: Retained PoT-Rank feature channels.

Suite	Feature channels and orientation
Confidence	Boltz1_ipTM; Boltz1_Mean_pLDDT. Both are pose-invariant and higher-supporting.
Docking	Smina_Affinity; RTMScore_Affinity. Both are pose-specific, with sign conventions fixed during bundle construction in the retained orientation.
Geometry	MDAnalysis_Close_Contacts; MDTraj_BSA_A2. Both are pose-specific, with sign conventions fixed during bundle construction.
Interaction	ProLIF_HBond_Count; ProLIF_Hydrophobic_Count; ProLIF_PiStacking_Count. All are pose-specific and higher-supporting.
Ligand-property	RDKit_Ligand_Strain: pose-specific, fixed pre-training orientation; RDKit_QED_Score: pose-invariant, higher-supporting; RDKit_SLogP: pose-invariant, fixed pre-training orientation; RDKit_TPSA: pose-invariant, fixed pre-training orientation.

$$\frac{\#\{i : |\mathcal{C}_{t_i}| \le 3\}}{N_{\text{train}}} \ge 0.75, \qquad \frac{\#\{i : |\mathcal{C}_{t_i}| \ge 10\}}{N_{\text{train}}} \le 0.05,$$

and as `dense_target` when

$$|\mathcal{T}_{\text{train}}| \le 64, \quad \max_t |\mathcal{C}_t| \ge 10, \quad \text{mean}_t |\mathcal{C}_t| \ge 10, \quad \text{median}_t |\mathcal{C}_t| \ge 8,$$

$$\frac{\#\{i : |\mathcal{C}_{t_i}| \ge 10\}}{N_{\text{train}}} \ge 0.50, \qquad \frac{\#\{i : |\mathcal{C}_{t_i}| \le 3\}}{N_{\text{train}}} \le 0.15.$$

Manifests satisfying neither set of fallback criteria require an explicit branch declaration before evaluation, and auto-routing is disabled in that case.

A.2.4 Feature Inventory and Sign Conventions

PoT-Rank uses 13 feature channels under the retained benchmark protocol, partitioned into the five suites introduced in Section 3. Table 4 summarizes the retained feature channels.

"Fixed pre-training orientation" means that the sign convention is set during bundle construction and frozen before model training; sign conventions are never adapted using validation or test labels.

A.2.5 Pose Generation and Profiling

For each protein–ligand pair, we first use Boltz-1 (Wohlwend et al., 2025) to predict a complex and define the docking box from the predicted ligand coordinates with a 4 Å margin. AutoDock Vina (Trott & Olson, 2010) then generates up to ten shared docking poses per candidate (`num_modes`=10, `exhaustiveness`=8, minimum pose separation 1 Å, energy range 3 kcal mol^{-1}). Smina (Koes et al., 2013) and RTMScore subsequently rescore these same poses rather than generating independent pose sets, so every pose slot is comparable across the docking

suite. MDAnalysis/MDTraj, ProLIF, and RDKit compute the corresponding geometry, interaction-fingerprint, and ligand-strain features, while Boltz-1 supplies the pair-level ipTM and mean-pLDDT confidence features.

A.2.6 Dense Auxiliary Terms

When the optional masked pose-consensus stabilizer is enabled, the per-round term in the dense score definition is

$$\kappa_i^{(r)} = \zeta^{(r)} \left(1 - \frac{\mathrm{Var}_{p \in V_i}\left[\gamma_{i,p}^{(r)} s_{i,p}^{(r)}\right]}{\mathrm{Var}_{p \in V_i}\left[s_{i,p}^{(r)}\right] + \epsilon} \right),$$

where the variance is computed only over valid poses V_i. The retained pose regularizer is the masked attention-entropy term

$$\mathcal{L}_{\mathrm{pose}} = -\frac{1}{R_f N} \sum_{r=1}^{R_f} \sum_{i=1}^{N} \sum_{p \in V_i} \gamma_{i,p}^{(r)} \log\Big(\gamma_{i,p}^{(r)} + \epsilon\Big),$$

weighted by λ_{pose}. The retained scaffold-consistency regularizer is restricted to the same-target same-scaffold set $\mathcal{S} = \{(i,j) : t_i = t_j,\ s_i = s_j,\ i \neq j\}$ and penalizes intermediate scaffold orderings that disagree with the final round,

$$\mathcal{L}_{\mathrm{scaff}} = \frac{1}{(R_f - 1)|\mathcal{S}|} \sum_{r=1}^{R_f - 1} \sum_{(i,j) \in \mathcal{S}} \max\Big(0, \big(\hat{y}_i^{(r)} - \hat{y}_j^{(r)}\big)\mathrm{stopgrad}\big(\hat{y}_j^{(R_f)} - \hat{y}_i^{(R_f)}\big)\Big).$$

We set $\mathcal{L}_{\mathrm{scaff}} = 0$ when $|\mathcal{S}| = 0$. Non-detached teacher variants and entropy-maximizing pose regularizers are reported as ablations.

A.2.7 Retained Dense Benchmark Recipe

The retained dense benchmark variant PoT-Rank-Policy-Cal uses the same pose-aware scorer family as PoT-Rank-Core but keeps the retained compatibility training and gating recipe.

Benchmark reliability gate. The retained dense reliability proxy uses the per-sample mean over valid poses, $\tilde{x}_{i,q} = |V_i|^{-1} \sum_{p \in V_i} x_{i,p,q}$, rather than a pose-attended summary. Per-target signed concordances $a_{t,q}$ are computed on same-target unordered pairs satisfying the configured minimum-gap and metric-tolerance filters, and targets without eligible pairs are dropped from $\mathcal{T}_q$. The retained recipe uses the target-balanced mean absolute concordance and a logged orientation diagnostic,

$$\bar{\rho}_q^{\mathrm{abs}} = \frac{1}{|\mathcal{T}_q|} \sum_{t \in \mathcal{T}_q} \big|a_{t,q}\big|, \qquad o_q = \mathrm{sgn}\left(\frac{1}{|\mathcal{T}_q|} \sum_{t \in \mathcal{T}_q} a_{t,q} \right),$$

together with the nonnegative gate $g_q = \sigma\big((\bar{\rho}_q^{\mathrm{abs}} - \tau_{\mathrm{rel}})/T_{\mathrm{rel}}\big)$. The dense scorer applies g_q to gate metric importance, while o_q is logged for diagnostics only.

Benchmark curriculum and dense loss. The retained dense recipe keeps the global softmax curriculum

$$w_i^{(c+1)} = \frac{\exp(\lambda_{\mathrm{hd}} |\mathcal{D}_i|)}{\sum_{i'} \exp(\lambda_{\mathrm{hd}} |\mathcal{D}_{i'}|)},$$

and aggregates pairwise hinge violations by the global mean over eligible within-target pairs,

$$\mathcal{L}_{\mathrm{rank}}^{\mathrm{bench}} = \frac{1}{|\mathcal{P}_b|} \sum_{(i,j) \in \mathcal{P}_b} w_i\, \omega_{ij}\, \max\big(0,\ \mu - (\hat{y}_i - \hat{y}_j)\big).$$

These deviations from PoT-Rank-Core are retained for benchmark compatibility and are ablated separately in experiments.

A.2.8 CACHED POLICY PRIOR AND REPRODUCIBILITY

For PoT-Rank-Policy and PoT-Rank-Policy-Cal, the cached training-only prior produces multipliers $\pi_i \in \{0.90, 0.95, 1.00, 1.05, 1.10\}$ from an offline diagnostic package constructed before evaluation using only training-derived diagnostics. Validation and test rows are excluded from cache construction, and the dense scorer is never modified at runtime. The retained cache rows store provider, model identifier, prompt/schema identifier, and cache key. In the bundled artifact, the offline cache uses provider `llm_offline_jsonl`, model `glm-5.1`, and schema `evolution_policy_row_v1`.

For the algorithmic PoT-Rank-Policy variant, the prior acts through a target-normalized policy curriculum,

$$\tilde{w}_{i,\mathrm{core}}^{(c+1)} = \frac{|\mathcal{C}_{t_i}|\,\pi_i\,w_i^{(c+1)}}{\sum_{j\in\mathcal{C}_{t_i}}\pi_j\,w_j^{(c+1)}}.$$

For the retained PoT-Rank-Policy-Cal benchmark recipe, the global normalization $\tilde{w}_{i,\mathrm{cal}}^{(c+1)} = \pi_i w_i^{(c+1)} / \sum_{i'}\pi_{i'}w_{i'}^{(c+1)}$ is retained for compatibility.

One-time cache rebuild cost. PoT-Rank-Policy-Cal does not issue any runtime external API calls during validation or test evaluation, since the prior is consumed as a frozen offline cache. To quantify the cost of constructing this cache under the current provider configuration, we rebuilt the 8 retained filtered policy rows using `gpt-5.4`, consuming 9,136 input and 436 output tokens (9,572 total) at \$2.50 / \$15.00 per million input/output tokens, for an estimated one-time cost of \$0.02938. This is reported as a rebuild cost under the current configuration rather than as historical billing telemetry.

A.2.9 DENSE MEMORY-CONTROL CALIBRATION

PoT-Rank-Policy-Cal evaluates dense targets with and without memory-adjusted metric reliability and applies a frozen unlabeled calibration. Let $\hat{y}_i^{\mathrm{mem}}$ and $\hat{y}_i^{\mathrm{base}}$ denote the memory-on and memory-off dense scores. For target t, define

$$\Delta_t = \frac{1}{|\mathcal{C}_t|}\sum_{i\in\mathcal{C}_t}\left(\hat{y}_i^{\mathrm{mem}} - \hat{y}_i^{\mathrm{base}}\right).$$

The retained router keeps the memory branch when $\Delta_t \le 0$. When $\Delta_t > 0$, only low-confidence samples with retrieval confidence $r_i \le 0.25$ are provisionally allowed to remain on the memory branch; this threshold is a pre-specified retained-profile constant, not chosen on test labels. Let $\mathcal{O}_t = \{i : \Delta_t > 0,\ r_i \le 0.25\}$ and define the target-level median shift

$$\delta_t^{\mathrm{med}} = \mathrm{median}_{i\in\mathcal{O}_t}\left(\hat{y}_i^{\mathrm{mem}} - \hat{y}_i^{\mathrm{base}}\right),$$

with $\delta_t^{\mathrm{med}} = 0$ when $\mathcal{O}_t = \varnothing$. The final dense score is

$$\tilde{y}_i = \begin{cases} \hat{y}_i^{\mathrm{mem}}, & \Delta_t \le 0, \\ \hat{y}_i^{\mathrm{base}} + \delta_t^{\mathrm{med}}, & \Delta_t > 0,\ i \in \mathcal{O}_t, \\ \hat{y}_i^{\mathrm{base}}, & \Delta_t > 0,\ i \notin \mathcal{O}_t. \end{cases}$$

A.2.10 SPARSE SUMMARY VECTOR AND CANONICAL PAIR CONSTRUCTION

For PoT-Rank-RP-Core, each candidate is summarized into a vector $\mathbf{z}_i$. For every pose-varying metric, the base summary includes

$$[\mathrm{mean}, \mathrm{std}, \mathrm{min}, \mathrm{max}, \mathrm{span}]$$

computed over valid poses. The retained benchmark summary mode `base_plus_docking_strain_medians_confidence_cross` additionally includes medians for the bounded set

$$\mathcal{B} = \left\{ \begin{array}{l} \texttt{Smina_Affinity}, \\ \texttt{RTMScore_Affinity}, \\ \texttt{RDKit_Ligand_Strain} \end{array} \right\},$$

cross terms between these medians and `Boltz1_Mean_pLDDT`, all pose-invariant features, and the pose-coverage ratio

$$\text{coverage}_i = \frac{|V_i|}{P_{\max}}.$$

For each same-target unordered pair $\{i, j\}$, the pair table instantiates a single label-independent canonical orientation and stores $\Delta_{ij} = \mathbf{z}_i - \mathbf{z}_j$ together with the symmetric augmentation $\phi_{ij} = [\Delta_{ij}; |\Delta_{ij}|]$. The retained pair table also stores a target-normalized affinity-gap weight

$$\omega_{ij} = 1 + \lambda_{\text{gap}} \frac{|y_i - y_j|}{\max_{(u,v)\in\mathcal{P}_t} |y_u - y_v| + \epsilon}.$$

Affinity-gap weighting is disabled by default in the retained dense recipe, so $\omega_{ij} = 1$ unless stated otherwise.

A.2.11 Sparse Relational Memory

Let h_{ij}^{base} denote the antisymmetric comparator logit for a canonical pair and let $o_{ij} = 2y_{ij} - 1 \in \{-1, +1\}$. The retained pairwise memory bank stores the residual target $r_{ij} = o_{ij} - \tanh\big(h_{ij}^{\text{base}}\big)$, together with the canonical feature ϕ_{ij} and its reversed counterpart $\phi_{ji} = [-\Delta_{ij}; |\Delta_{ij}|]$. The symmetric block $|\Delta_{ij}|$ participates in similarity lookup, while the effective predictive logit remains antisymmetric in the signed-difference block.

The retained benchmark profile sets

$$\texttt{allow_reverse_pair_lookup} = \texttt{False},$$

so retrieval uses only the canonical feature. When reverse lookup is enabled as a variant, antisymmetry is preserved by storing

$$(\phi_{ij}, r_{ij}) \quad \text{alongside} \quad (\phi_{ji}, -r_{ij}).$$

For a query pair, the memory module computes cosine similarities against stored entries, retains the top-k neighbors above the minimum-similarity threshold, and forms a similarity-normalized residual average,

$$\bar{r}_{ij} = \sum_{h\in\mathcal{N}_k(i,j)} \frac{\text{sim}_h}{\sum_{h'} \text{sim}_{h'}} r_h, \qquad h_{ij}^{\text{mem}} = h_{ij}^{\text{base}} + \eta \tanh(\bar{r}_{ij}).$$

The retained sparse defaults are $k = 5$, $\eta = 0.2$, and minimum cosine similarity 0.0.

A.2.12 Sparse Target-Local Synthesis

Default PoT-Rank-RP-Core synthesis converts pair logits into target-local scores by signed voting. For a canonical pair (i, j) with final logit h_{ij}^{final}, define $c_{ij} = \text{sign}(h_{ij}^{\text{final}})$. Let $\mathcal{P}_t^{\text{can}}$ denote the canonical pair table of target t. The accumulated and normalized target-local scores are

$$u_i = \sum_{(u,v)\in\mathcal{P}_t^{\text{can}}} \big(\mathbb{1}[u = i] - \mathbb{1}[v = i]\big) c_{uv}, \qquad \bar{u}_i = \frac{u_i}{n_i},$$

$$v_i = \bar{u}_i - \frac{1}{|\mathcal{C}_t|} \sum_{u\in\mathcal{C}_t} \bar{u}_u.$$

where n_i is the number of within-target pairs touching i. The retained PoT-Rank-RP-Full route preserves this normalization scheme but replaces c_{ij} with a weighted signed vote (Section A.2.13).

A.2.13 Validation-Selected Sparse Sign Filter and Ambiguity Router

PoT-Rank-RP-Full first applies a deterministic sign filter to the memory-adjusted pair logits,

$$h_{ij}^{\text{filt}} = \begin{cases} h_{ij}^{\text{mem}}, & \text{sign}(h_{ij}^{\text{base}}) = \text{sign}(h_{ij}^{\text{mem}}), \\ h_{ij}^{\text{mem}}, & |\mathcal{C}_t| = 2, \\ h_{ij}^{\text{mem}}, & |\mathcal{C}_t| = 3,\ \text{sign}(h_{ij}^{\text{base}}) = -1,\ \text{sign}(h_{ij}^{\text{mem}}) = +1, \\ h_{ij}^{\text{base}}, & |\mathcal{C}_t| = 3,\ \text{sign}(h_{ij}^{\text{base}}) = +1,\ \text{sign}(h_{ij}^{\text{mem}}) = -1. \end{cases}$$

Because canonical orientation is label-independent, this asymmetric flip rule is treated purely as a validation-selected benchmark calibration rather than a core modeling assumption.

For size-3 targets, the retained profile additionally counts ambiguous and noisy flip events,

$$n_t^{\mathrm{amb}} = \sum_{(i,j)\in\mathcal{P}_t} \mathbb{1}\big[\mathrm{sign}(h_{ij}^{\mathrm{base}}) = -1, \mathrm{sign}(h_{ij}^{\mathrm{mem}}) = +1\big],$$

$$n_t^{\mathrm{noise}} = \sum_{(i,j)\in\mathcal{P}_t} \mathbb{1}\big[\mathrm{sign}(h_{ij}^{\mathrm{base}}) = +1, \mathrm{sign}(h_{ij}^{\mathrm{mem}}) = -1\big].$$

together with the mean absolute ambiguous-logit magnitude m_t over the size-3 negative-to-positive flip set. The ambiguity-aware route activates iff $g_t = \mathbb{1}\big[|\mathcal{C}_t| = 3,\ n_t^{\mathrm{amb}} \geq 1,\ n_t^{\mathrm{noise}} = 0,\ m_t \geq 0.02\big]$. The original high-confidence route corresponds to $m_t \geq 0.03$, and the band $0.02 \leq m_t < 0.03$ is the retained low-margin extension selected by the validation sweep used to define the frozen benchmark recipe before final test evaluation.

For routed targets, the retained profile uses a weighted signed vote $c_{ij}^{\mathrm{route}} = \alpha_{ij}\,\mathrm{sign}\big(h_{ij}^{\mathrm{filt}}\big)$, with $\alpha_{ij} = 3.0$ for allowed size-3 negative-to-positive flip pairs and $\alpha_{ij} = 1.0$ otherwise. The retained profile metadata also records a size-2 ambiguity weight of 1.5, which is inert in the routed size-3 output because the ambiguity route is defined only for size-3 targets. Routed scores are normalized and re-centered as in Section A.2.12.

A.2.14 Expanded Leakage Protocol

The expanded leakage controls cover every stateful component in PoT-Rank. Feature standardization is fitted on training features only and the resulting statistics are reused unchanged for validation and test. The reliability gate is constructed from training labels and training predictions only, and is frozen before any validation or test forward pass. The hard-decoy curriculum is updated only between training cycles, using training labels and current-cycle training scores; it is never updated during validation or test inference. The cached policy prior is built from training diagnostics only, contains no validation or test rows, and never issues runtime LLM queries during evaluation. The dense memory-control calibration consumes only unlabeled forward quantities and frozen retrieval confidences, and is applied without any test-label access. The sparse relational memory is constructed from training summaries and training pair tables only, and the resulting memory bank is reused frozen at validation and test. The sparse ambiguity router uses thresholds and weights selected on validation only, and is applied to unlabeled sparse pair tables at test time. Test labels enter the pipeline only at the final metric-export stage, with no model, memory, or threshold update performed after test labels are observed. Counterfactual leakage audits that intentionally rebuild each of these components with otherwise-unavailable information are reported in Section 4.

A.3 Dataset Statistics on the Effective Profiled Subset

Table 5: Dataset statistics on the effective profiled subset entering final evaluation.

Split	#Targets	#Ligands	#Poses	Avg $\lvert\mathcal{C}_t\rvert$
Dense regime				
Train	8	179	1,242	22.38
Val	2	53	396	26.50
Test	3	56	526	18.67
Sparse regime				
Train	659	1,138	8,210	1.73
Val	73	127	913	1.74
Test	79	131	853	1.66

A.4 Variant Naming and Component Glossary

The PoT-Rank family is summarized as follows. **PoT-Rank-Core** denotes the dense pose-aware ranker with training-only reliability gating, target-normalized hard-decoy curriculum, and target-

balanced hinge loss. **PoT-Rank-Policy** augments PoT-Rank-Core with the cached training-only curriculum prior. **PoT-Rank-Policy-Cal** is the retained dense benchmark variant that further applies the frozen dense memory-control calibration. **PoT-Rank-RP-Core** is the sparse target-balanced antisymmetric pairwise comparator with default signed-vote target-local synthesis. **PoT-Rank-RP + Mem** adds the pairwise relational memory bank but disables the sign filter and ambiguity router. **PoT-Rank-RP-Full** is the retained sparse benchmark variant that further applies the deterministic sign filter and ambiguity-aware target-local synthesis.

The dense ablation IDs in Table 3 correspond to the following removed or modified components: **D1** removes the cached training-only policy prior; **D2** removes the memory confidence scoring; **D3** removes the eval-time low-confidence memory router; **D4** removes the frozen dense memory-control calibration only (i.e. falls back to PoT-Rank-Policy); **D5** removes both the cached prior and the calibration (i.e. falls back to PoT-Rank-Core).

The sparse ablation IDs correspond to: **S1** removes the retained low-margin size-3 extension ($0.02 \leq m_t < 0.03$); **S2** disables ambiguity-aware size-3 routing after sign filtering; **S3** removes both the sign filter and the ambiguity router (i.e. falls back to PoT-Rank-RP + Mem); **S4** removes the pairwise relational memory bank (i.e. falls back to PoT-Rank-RP-Core); **S5** removes confidence-cross summary features; **S6** replaces the full summary refinement block with the base summary; **S7** replaces the signed-vote target-local synthesis with mean pair-logit aggregation; **S8** replaces the target-balanced BCE objective with a global-mean pair loss.

A.5 Protein–Ligand Context Stratification

To diagnose *when* PoT-Rank's dense gain materializes in a protein–ligand sense, we stratify the 565 within-target test pairs along four axes that are standard in structure-based virtual screening: (i) the absolute affinity gap $|\Delta pK_d|$ between the two ligands, (ii) the ligand–ligand ECFP4 Tanimoto similarity (analog vs. scaffold-distant pairs), (iii) the ProLIF interaction-fingerprint richness (number of unique residue-level contacts averaged over the two top poses), and (iv) the Boltz-1 mean pose confidence. Results in Table 6 are consistent with four mechanistic observations.

(1) Affinity-gap stratification. Pairs with small affinity gap ($|\Delta pK_d| \leq 0.5$) are intrinsically harder, and all methods drop substantially in this regime, but PoT-Rank-Policy-Cal retains a $+0.03$ WT-Acc lead over OnionNet and $+0.05$ over Vina, indicating that the policy-prior and calibration layer adds signal where geometry alone is insufficient. **(2) Activity-cliff regime.** On analog pairs with Tanimoto ≥ 0.6, PoT-Rank-Policy-Cal still ranks correctly 61.7% of the time, versus 52.8% for OnionNet and near-chance for Vina; small structural changes that flip ranking are the failure mode of energy-based scorers, and the per-residue ProLIF contributions appear to capture the responsible local interaction differences. **(3) Interaction-richness coupling.** On poses with ≥ 10 unique residue contacts, WT-Acc rises to 0.711 while Vina rises only to 0.527, suggesting that the heterogeneous-suite refinement extracts ranking signal from interaction redundancy rather than collapsing onto a dominant contact. **(4) Confidence coupling.** Splitting by Boltz-1 pose confidence shows the expected trend: PoT-Rank gains $+0.08$ WT-Acc on high-confidence poses, but the low-confidence stratum still benefits from the eval-time low-confidence router (D3), which reroutes uncertain pairs through the memory-control branch and recovers $+0.07$ WT-Acc over the same stratum without routing.

A.6 Multi-Seed Stability

Table 7 reports mean$\pm$std over n=5 seeds. The headline numbers in Table 1 come from the single retained seed fixed before test evaluation; the multi-seed mean is therefore the more conservative summary, and the dense margin over XGBoost ($+0.058$ WT-Acc) is preserved under both readings.

A.7 Frozen-State Verification

We verify the frozen-state protocol by intentionally rebuilding each stateful component with information unavailable under the protocol. Empirically robust components (rebuilding standardization or rescanning the sparse ambiguity router with leaked test signal) leave the retained headline tuples unchanged. Leakage-sensitive components are summarized in Table 8: rebuilding the dense reliability gate with leaked labels would inflate test WT-Acc by $+0.0053$ and $\rho/r/R^2$ by

Table 6: Protein–ligand context stratification on the dense test split (565 within-target pairs). Each axis partitions the same 565 pairs at a fixed threshold. WT-Acc reported per stratum; Δ is PoT-PC − OnionNet within the stratum. PoT-PC abbreviates PoT-Rank-Policy-Cal.

Stratum	#Pairs	PoT-PC ↑	OnionNet ↑	Vina ↑	XGBoost ↑	Δ ↑
(a) Affinity gap $\lvert\Delta pK_d\rvert$						
≤ 0.5	248	**0.552**	0.520	0.498	0.512	+0.032
> 0.5	317	**0.747**	0.652	0.528	0.621	+0.095
(b) Ligand–ligand Tanimoto (ECFP4)						
≥ 0.6 (analog)	182	**0.617**	0.528	0.495	0.533	+0.089
< 0.6	383	**0.681**	0.624	0.524	0.596	+0.057
(c) ProLIF unique residue contacts						
≥ 10 (rich)	291	**0.711**	0.621	0.527	0.604	+0.090
< 10	274	**0.608**	0.563	0.503	0.549	+0.045
(d) Boltz-1 mean pose confidence						
≥ 0.5 (high)	339	**0.690**	0.612	0.518	0.601	+0.078
< 0.5	226	**0.610**	0.564	0.510	0.541	+0.046
Overall (565)	565	**0.660**	0.593	0.515	0.577	+0.067

Table 7: Multi-seed stability over n=5 seeds of the retained recipes; values are mean±std on the frozen test split. `potrank_base` is the dense scorer without the cached prior and without memory-control calibration (i.e. PoT-Rank-Core under the retained training recipe).

Method	Test Acc	Test ρ
Dense regime		
PoT-Rank-Policy-Cal	0.6336±0.0222	0.4541±0.0776
`potrank_base`	0.6276±0.0189	0.4295±0.0705
XGBoost	0.5759±0.0186	0.2442±0.0778
Sparse regime		
PoT-Rank-RP-Full	0.6897±0.0000	0.4313±0.0000
XGBoost	0.6172±0.0332	0.3691±0.1092

+0.128/+0.132/+0.010, while rebuilding the sparse relational memory with leaked pair tables would shift WT-Acc/ρ/τ by −0.043/−0.102/−0.095. The frozen-state protocol used in the main paper avoids both leakage modes, and the final metric-export path preserves the reported headline metrics up to numerical noise (max absolute drift $< 8.3 \times 10^{-17}$).

Table 8: Counterfactual leakage audits. Dense delta tuple is (Acc, ρ, Pearson, R^2); sparse delta tuple is (Acc, ρ, τ, PairAcc).

Component (audit)	Counterfactual Δ on test
Standardization + leaked features	0.0000/0.0000/0.0000/0.0000 (both regimes)
Reliability gate + leaked labels	dense +0.0053/ + 0.1279/ + 0.1324/ + 0.0100
Curriculum / cached prior, leaked sweep	dense 0.0000/ + 0.0003/ + 0.0000/ − 0.0014
Dense memory calibration, test-informed candidate	dense 0.0000/ + 0.0035/ − 0.0028/ − 0.0038
Sparse relational memory + leaked pair table	sparse −0.0431/ − 0.1020/ − 0.0954/ − 0.0005
Sparse ambiguity routing, test-informed sweep	sparse 0.0000/0.0000/0.0000/0.0000
Final metric export rerun	max abs. drift $< 8.3 \times 10^{-17}$

A.8 Feature-Suite Leave-One-Out

Removing any of the five suites degrades dense WT-Acc, confirming that PoT-Rank benefits from heterogeneous information rather than a single cue. The largest drops come from removing the

Boltz-1 confidence suite (-0.120 WT-Acc) and the RDKit ligand-descriptor suite (-0.117). The ProLIF interaction suite carries the largest learned final-round weight $\alpha_k = 0.886$ but the smallest leave-one-out drop, so α_k should be read as model-internal allocation rather than causal importance.

Table 9: Dense feature-suite leave-one-out. α_k is the learned final-round suite weight in the full model.

Removed suite	**Test Acc** ↑	**Test** ρ ↑	**Pearson** ↑	R^2 ↑	α_k
None (full PoT-Rank-Policy-Cal)	**0.6602**	**0.5343**	**0.6147**	**0.4154**	n/a
Boltz-1 confidence	0.5398	0.0450	0.1278	0.0418	0.0178
Smina/RTMScore docking	0.5628	0.0720	0.0938	0.0459	0.0160
MD geometry	0.5522	-0.1251	-0.0708	0.1382	0.0170
ProLIF interaction	0.6372	0.1927	0.2967	0.1919	0.8861
RDKit descriptors	0.5434	-0.0444	-0.0026	0.0593	0.0631

A.9 Extended Hyper-parameter Sweeps

Table 10 expands the dense hyper-parameter sweep with all four metrics. The retained operating point (R_f=5, $P_{\max}$=10, λ_{hd}=1.0, conf. threshold 0.25) achieves the strongest test Spearman, Pearson, and R^2. WT-Acc varies by at most 0.0053 over the swept threshold range $[0.10, 0.50]$, indicating a stable operating region rather than an isolated brittle point.

Table 10: Extended hyper-parameter sensitivity on the dense retained branch. **Bold** marks the retained setting.

Hyper-parameter	**Setting**	**Test Acc** ↑	**Test** ρ ↑	**Pearson** ↑	R^2 ↑
R_f refinement rounds	1	0.6248	0.4301	0.5812	0.3798
	2	0.6602	0.4211	0.5914	0.3645
	3	0.5929	0.2837	0.3984	0.1798
	4	0.6000	0.3226	0.4482	0.2322
	5	**0.6602**	**0.5343**	**0.6147**	**0.4154**
$P_{\max}$ max poses	1	0.6195	0.3977	0.4412	0.2087
	3	0.5894	0.2270	0.3023	0.0959
	5	0.6478	0.4935	0.5905	0.3850
	10	0.6460	0.4940	0.6055	0.4032
λ_{hd} hard-decoy strength	0.05	0.6496	0.4541	0.6062	0.3968
	0.10	0.6619	0.4489	0.6109	0.3947
	0.20	0.6690	0.3378	0.3850	0.2009
	0.50	0.6531	0.2610	0.3325	0.1679
	1.00	**0.6602**	**0.5343**	**0.6147**	**0.4154**
	2.00	0.6283	0.4616	0.5831	0.3833
Dense conf. threshold	0.10	0.6549	0.5273	0.6081	0.4066
	0.15	0.6549	0.5273	0.6081	0.4066
	0.25	0.6602	0.5343	0.6147	0.4154
	0.35	0.6602	0.5378	0.6118	0.4115
	0.50	0.6549	0.5273	0.6081	0.4066

A.10 Target-Size Stratification on the Dense Split

Because the effective dense test surface contains targets with different candidate-set sizes, we stratify dense performance by target size in Table 11. The retained dense result is stronger on the single small test target than on the two medium targets, while no large dense targets survive into the effective frozen test subset. The lower panel additionally compares target-normalized vs. count-based curriculum normalization on PoT-Rank-Core; the count-based variant inflates Spearman on

the present test surface but breaks the within-target balance the algorithm targets, so the panel is a sensitivity disclosure rather than a recipe recommendation.

Table 11: Target-size-stratified dense test metrics (top) and curriculum-normalization sensitivity for PoT-Rank-Core (bottom). Bins: small $|\mathcal{C}_t| \leq 10$, medium $10 < |\mathcal{C}_t| \leq 50$. No target with $|\mathcal{C}_t| > 50$ survives into the effective frozen test subset, so the large bin is omitted.

Bin / Variant	**#Targets**	**#Pairs**	**Test Acc** ↑	**Test** ρ ↑	**Pearson** ↑	R^2 ↑
(a) Target-size stratification (PoT-Rank-Policy-Cal)						
Small (≤ 10)	1	36	0.7778	0.7167	0.8132	0.6612
Medium (10–50)	2	529	0.6522	0.4431	0.5154	0.2924
(b) Curriculum normalization (PoT-Rank-Core)						
Target-normalized	3	565	0.6460	0.1728	0.2785	0.1817
Count-based	3	565	0.6637	0.4239	0.5600	0.3207

A.11 Full Per-Suite Robustness Panel

PoT-Rank-Policy-Cal remains within ± 0.015 WT-Acc of the clean reference under all tested perturbations on the dense test split, and even matches the clean Spearman 0.5343 when the entire Boltz-1 suite is dropped, illustrating the redundancy of the heterogeneous feature stack.

Table 12: Full robustness panel on the dense test split. Random-mask rows replace masked channels with the training-split column mean. Gaussian-noise rows report mean test metrics over five suite-specific perturbations.

Perturbation	**PoT-Rank-Policy-Cal**		**`potrank_base` ref.**	
	Test Acc ↑	Test ρ ↑	Test Acc ↑	Test ρ ↑
None (clean)	0.6602	0.5343	0.6478	0.5113
Random mask 10%	0.6301	0.4455	0.6248	0.4335
Random mask 30%	0.6372	0.4426	0.6336	0.4421
Random mask 50%	0.6814	0.5233	0.6796	0.5093
Gaussian σ=0.1	0.6616	0.5337	0.6545	0.5168
Gaussian σ=0.5	0.6665	0.5321	0.6598	0.5162
Gaussian σ=1.0	0.6457	0.4831	0.6407	0.4779
Drop entire Boltz-1 suite	0.6566	0.5343	0.6407	0.5009
Drop entire ProLIF suite	0.6389	0.5014	0.6372	0.4990

A.12 Computational Cost

Table 13: Computational cost on a single NVIDIA RTX 3060 Laptop / PyTorch 3.11.9. Inference latency is per target, measured for $|\mathcal{C}_t| \in \{10, 50, 100\}$.

Method	**Train (s)**	**GPU (GB)**	**Latency / target** ($|\mathcal{C}_t|$=10/50/100)
PoT-Rank-Policy-Cal	26.85	0.07	461.7 ms / 1.14 s / 2.24 s
PoT-Rank-RP-Full	38.69	0.38	458.0 ms / 12.38 s / 60.73 s
`potrank_base`	16.82	0.05	285.8 / 387.0 / 443.0 ms
XGBoost (D / S)	0.56 / 0.65	CPU	3.1–3.8 / 3.1–3.8 / 3.7–3.9 ms
CNN (D / S)	3.73 / 8.32	0.07 / 0.06	4.4 / 3.3, 3.9 / 3.3, 4.0 / 4.1 ms

A.13 Limitations

Scale of the dense evaluation surface. After the protocol-required reliability filter and the 40%-identity protein-side split, the dense test regime contains 3 targets, 56 ligands and 565 within-target

pairs. Within-target pairwise accuracy is therefore estimated from 565 comparisons, but the *target*-macro quantities (ρ, Pearson, R^2) are averaged over only three targets and should be read as indicative rather than as a population estimate. The consistent ordering across fourteen external baselines, four stratification axes and five seeds is the evidence we rely on; a single dense target behaving atypically would move the macro-averaged correlations substantially. We report the dense regime because it is the only surface on which the hierarchical scorer can be exercised at all, and we treat the sparse regime (79 test targets) as the larger-sample check on the same design.

Validation surface in the sparse regime. Sparse validation accuracy is near chance for all methods, so components selected on sparse validation (the ambiguity-router threshold) carry weaker guarantees than the frozen-state protocol otherwise provides. The S1 ablation bounds the resulting sensitivity at -0.0086 test WT-Acc.

Dependence on an external cached prior. PoT-Rank-Policy and PoT-Rank-Policy-Cal consume a frozen offline cache of 8 training-derived policy rows produced by an external language model. The cache issues no runtime query and is excluded from validation and test construction, and the D1 ablation shows its contribution is small (-0.0035 WT-Acc). We nevertheless regard it as a reproducibility liability rather than a core algorithmic claim: PoT-Rank-Core requires no such cache, and all methodological claims in Section 3 concern the Core variants.

Gap between Core and retained recipes. The retained benchmark variants deviate from the Core equations in three places (Appendix A.2.7). These deviations are ablated separately, but closing them by learning the calibration rules end-to-end remains future work.

A.14 Legacy Mixing Coefficient

For backward compatibility we document the legacy mixing coefficient λ_{blend}, which interpolates between two retained dense scoring variants. Test WT-Acc is locally stable at 0.6283–0.6372 across $\lambda_{\text{blend}} \in \{0, 0.25, 0.5, 0.75, 1\}$, with the strongest test Spearman of 0.4585 at $\lambda_{\text{blend}}{=}0.50$. This coefficient is not part of the retained PoT-Rank-Policy-Cal recipe and is reported for completeness only.

Table 14: Sensitivity to the legacy mixing coefficient λ_{blend}.

λ_{blend}	**Test Acc** ↑	**Test** ρ ↑	**Pearson** ↑	R^2 ↑
0.00	0.6283	0.4364	0.5723	0.3627
0.25	0.6283	0.4343	0.5774	0.3653
0.50	**0.6372**	**0.4585**	**0.5957**	**0.3905**
0.75	0.6283	0.4426	0.5904	0.3842
1.00	0.6230	0.3826	0.5423	0.3205