BEYOND STRUCTURE GENERATION: CO-FOLDING REPRESENTATIONS FOR BINDING AFFINITY AND ADMET PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Binding affinity and enzyme-dependent metabolism are distinct molecular outcomes that both involve protein–ligand recognition. We introduce PairSense, a framework for predicting binding affinity and CYP substrate status from protein-conditioned co-folding representations without explicit structure generation. PairSense combines Boltz-2's Pairformer embeddings with predicted pairwise distance distributions, retaining probabilistic geometric information while replacing distances derived from diffusion-generated coordinates. With the co-folding trunk frozen, we fine-tune only the affinity module on public data curated to remove benchmark overlap. PairSense improves over the original Boltz-2 by approximately 0.09 in Pearson correlation on the four-target FEP+ benchmark and 0.16 in Spearman correlation on OpenBind, with additional gains on CASP16 and OpenFE876. Keeping the affinity-adapted interaction representation fixed, we jointly fine-tune its native binary readout with molecular fingerprints on three cytochrome P450 substrate tasks. The joint model achieves higher mean scores than the published TDC leaderboard references. Controlled comparisons show that additional binding supervision improves CYP prediction and that protein context adds value beyond ligand features and task identity. These findings support co-folding representations as a reusable basis for predicting binding affinity and protein-mediated molecular properties beyond structure generation.

1 INTRODUCTION

Predicting small-molecule properties is essential to identifying compounds with suitable activity, pharmacokinetics, and safety. Many of these properties arise from interactions between a molecule and its biological environment. Binding affinity describes association with a target, metabolism involves recognition by enzymes, and transport and certain toxicity endpoints depend on interactions with transporters or other proteins. Although commonly recorded as attributes of individual compounds, these outcomes depend on both the molecule and its biomolecular partners. This interaction-based view motivates representations that describe a compound in its relevant biological context and can support prediction across different drug-related properties.

Protein–ligand affinity prediction provides a familiar example of this view: many approaches explicitly incorporate both the protein and the ligand. In contrast, many ADMET predictors primarily describe the small molecule (Yang et al., 2019; Hu et al., 2020; Klaser et al., 2024), even when the endpoint concerns a specific protein. Standard CYP substrate benchmarks, for example, map a molecular structure to substrate status for a designated enzyme, leaving the enzyme implicit in the task definition (Huang et al., 2021). Molecular representations can learn associations between chemical features and these labels, but do not explicitly describe the corresponding protein–molecule interaction. Protein-aware approaches have begun to address this limitation in CYP prediction (Yang et al., 2026), motivating a broader question: how can representations of biomolecular interactions be used across molecular property tasks with identifiable protein contexts?

Recent co-folding models offer a promising foundation. These models jointly process proteins and ligands to predict complex structures, producing intermediate interaction representations and geometric predictions (Abramson et al., 2024; Wohlwend et al., 2024; Chai Discovery et al., 2024;

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.

Protenix Team et al., 2026). The information learned for structure prediction may also be useful for predicting the properties and biological outcomes of an interaction. Boltz-2 already connects co-folding representations to binding prediction, combining internal features with geometry derived from generated coordinates (Passaro et al., 2025). TerraBind directly uses learned structural representations and predicted distance-bin probabilities for affinity prediction (Rossi et al., 2026). Jang et al. (2026) further investigate co-folding representations for molecular property prediction, including the contribution of protein context. These developments motivate examining co-folding as a source of reusable interaction representations for both affinity and protein-dependent molecular properties.

Here, we introduce PairSense, a framework that predicts molecular properties from representations of a small molecule together with its relevant protein. PairSense combines interaction features from a frozen Boltz-2 backbone with predicted pairwise distance distributions, without generating explicit complex coordinates. A probability-weighted distance-bin embedding makes the predicted geometric distributions available to the downstream module. We further adapt the affinity module under binding supervision while keeping the co-folding backbone fixed. Controlled comparisons with point-estimate encodings and no explicit distogram input assess the contribution of this geometric interface.

We examine the framework through binding affinity and CYP substrate prediction. Affinity provides a direct test of interaction-strength prediction, while CYP substrate recognition extends the evaluation to an enzyme-dependent metabolic outcome. Binding affinity and substrate status are distinct quantities, but both involve recognition in a protein-specific environment. We therefore investigate whether the interaction representation and native prediction pathway used for affinity can provide a useful starting point for substrate prediction. This extension tests the value of the learned representations beyond their original binding endpoint.

For CYP prediction, each example pairs a candidate molecule with its corresponding enzyme. CYP2C9, CYP2D6, and CYP3A4 are evaluated through a shared protein-conditioned predictor, with enzyme context supplied through the protein input. This formulation allows related endpoints to share parameters and supervision while retaining their different biological contexts. We reuse the affinity-adapted interaction representation and fine-tune its native binary readout on substrate labels, with molecular fingerprints supplying complementary ligand features. Experiments on four affinity benchmarks and three CYP tasks assess predictive performance, representation reuse, and the contributions of explicit protein information and joint learning.

Our contributions comprise a representation framework and two findings about its use in molecular property prediction:

- **Co-folding interaction representations for molecular property prediction.** We introduce PairSense, a framework that represents a small molecule together with its relevant protein to predict drug-related properties. PairSense directly combines pretrained co-folding interaction features with predicted distance distributions, making information learned for complex structure prediction available to downstream predictors without explicit coordinate generation. We apply this framework to binding affinity and protein-dependent metabolism.

- **Reuse of interaction representations across binding and metabolic endpoints.** We demonstrate that the interaction representations used for affinity prediction can also support CYP substrate recognition, extending their application from binding strength to an enzyme-dependent metabolic outcome. Reusing the interaction representation and its native prediction pathway enables this extension, while controlled comparisons show that additional binding supervision improves subsequent CYP prediction across all three evaluated enzymes.

- **The value of explicit protein context for shared property learning.** We formulate substrate prediction across CYP enzymes through a shared protein-conditioned predictor, with the corresponding enzyme supplied as an input. Under joint training, native protein information improves prediction beyond ligand features and task identity alone. Joint learning also improves over separate task models across all three CYP endpoints, supporting the use of explicit interaction context and shared supervision for these related property tasks.

2 Related Work

Co-folding for binding affinity and geometric conditioning. Recent co-folding models have extended biomolecular structure prediction toward protein–ligand binding prediction. Boltz-2 builds an affinity module on its co-folding backbone, combining pair representations with geometry derived from diffusion-generated complex coordinates (Passaro et al., 2025). TerraBind instead conditions affinity prediction directly on a coarse structural Pairformer and predicted distance-bin probabilities; its separate pose module does not pass generated coordinates to the affinity module (Rossi et al., 2026). PairSense studies a related interface by reusing a frozen Boltz-2 backbone and comparing probabilistic, point-estimate, and no-explicit-distogram inputs under matched training conditions.

Co-folding representations for molecular property prediction. Jang et al. (2026) study co-folding representations from Boltz-2 for molecular property prediction. Their standard TDC protocol uses ligand-only inputs with external nonlinear probes, while additional experiments show that relevant protein context can benefit some enzyme-related endpoints. PairSense focuses on affinity-supervised adaptation of protein–ligand interaction representations and reuse of the affinity module's native binary prediction pathway for CYP substrate prediction.

Modeling enzyme context in CYP substrate prediction. CYP substrate predictors differ in how they represent enzyme context. Fang et al. (2024) jointly predict substrate status for five CYP isoforms using molecular fingerprints and graphs, with enzyme identity specified by the prediction task rather than an explicit protein representation. EviCYP incorporates CYP sequences alongside molecular features (Yang et al., 2026). PairSense jointly learns from molecule–enzyme pairs using co-folding interaction representations and a binary readout shared across the three CYP tasks, without enzyme-specific heads or additional task identifiers.

3 PairSense

PairSense represents each small molecule together with its relevant protein for binding affinity and CYP substrate prediction. PairSense keeps the Boltz-2 trunk frozen and never runs its structure generation module (Passaro et al., 2025). A single trunk forward pass per protein–ligand pair produces the inputs described in Section 3.1; these are cropped, cached, and reused by every subsequent training stage. Learning is confined to the affinity module. Using the probabilistic geometry interface (Section 3.2), we fine-tune this module for binding affinity (Section 3.3) and then transfer it to CYP substrate prediction through its native classification pathway (Section 3.4). Figure 1 summarizes the pipeline.

3.1 Pocket-cropped co-folding representations

A protein–ligand complex is tokenized into L tokens, one per protein residue and one per ligand heavy atom. We denote by $\mathcal{I}_{\text{lig}}$ the set of ligand tokens. A single forward pass of the frozen Boltz-2 trunk on the full-length input yields the pair representation $Z \in \mathbb{R}^{L\times L\times 128}$ and distogram logits $D \in \mathbb{R}^{L\times L\times 64}$. Separately, the input embedder produces $S_{\text{in}} \in \mathbb{R}^{L\times 384}$ (`s_inputs`). Applying a softmax over the last axis of D gives a categorical distribution $p_{ij} \in \Delta^{64}$ over 64 distance bins. The 63 bin boundaries are evenly spaced from 2 to 22 Å, defining 64 bins including two open-ended extremes; their centers are denoted $d_1, \ldots, d_{64}$.

Two decisions are made from these outputs, and we keep them separate. The *crop* decides which tokens the affinity module sees. Following the 15 Å pocket definition of TerraBind (Rossi et al., 2026), we compute $\mathbb{E}[d_{ij}] = \sum_b p_{ijb} d_b$ and keep a protein token i if $\min_{j\in\mathcal{I}_{\text{lig}}} \mathbb{E}[d_{ij}] < 15\,\text{Å}$; all ligand tokens are kept. For the retained pairs, the predicted distributions provide explicit geometric conditioning (Section 3.2). The crop is therefore a token-selection rule; it does not commit the model to a single structure. The retained set contains N tokens; we write the cropped arrays as $S_{\text{in},c}$, Z_c, and D_c.

PairSense runs the trunk on the full-length complex and then crops its representations using the trunk's own distance predictions. The retained $(S_{\text{in},c}, Z_c, D_c)$ triple is cached once per complex and reused during affinity or CYP training, with softmax applied to D_c as needed. This avoids repeated trunk computation and requires no diffusion-based structure generation. For comparison,

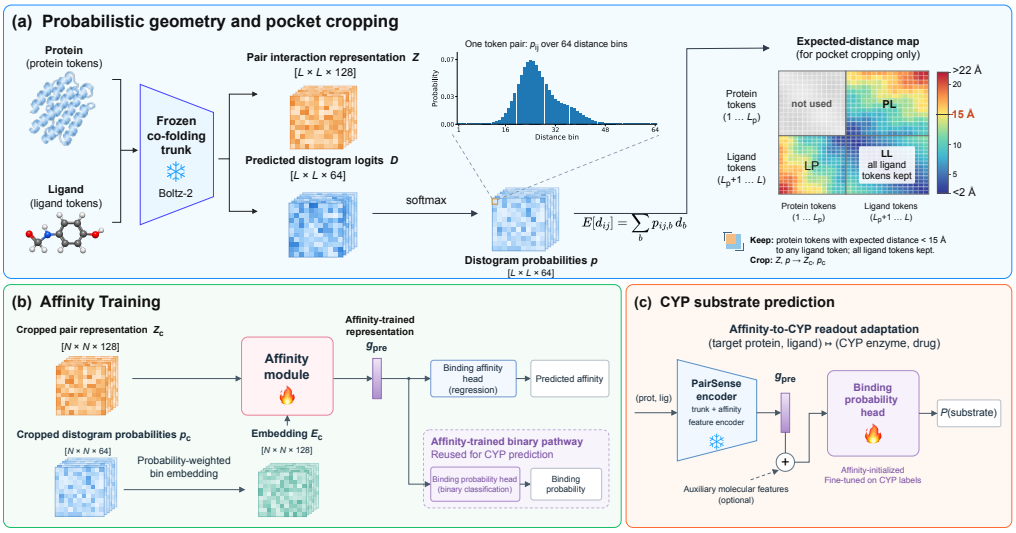

Figure 1: Overview of PairSense. (a) A frozen Boltz-2 trunk processes each full-length protein–ligand pair once, without diffusion-based structure generation; logits D are softmaxed to p, and expected distances select a 15 Å pocket while retaining all ligand tokens. (b) The cropped pair representation and probability-weighted distance embedding are fed to the fine-tuned affinity module, whose pooled output drives regression and binary heads. (c) The binary pathway is reused for CYP substrate prediction after joint adaptation, without adding a new prediction head. The single-token input features S_{in} are omitted from the figure for clarity.

Appendix B.1 describes the original Boltz-2 inference pipeline, including its diffusion-based cropping and multi-sample procedure.

3.2 Probabilistic Geometry Interface

PairSense uses the cropped distance distributions to condition the affinity module directly. In the original Boltz-2 pipeline, pairwise distances from a diffusion-generated structure are discretized into 64 bins, and each bin index selects a row from a learned distance-bin embedding matrix $E^{\text{hard}} \in \mathbb{R}^{64\times128}$, whose rows we denote $e_1, \ldots, e_{64}$. PairSense instead computes a probability-weighted combination of these embeddings:

$$E_{c,ij} = \sum_{b=1}^{64} p_{c,ijb}\, e_b, \tag{1}$$

which is implemented as a bias-free linear map $\mathbb{R}^{64} \to \mathbb{R}^{128}$ applied to $p_{c,ij}$. The map is initialized from the transposed distance-bin embedding matrix of the eight-layer member (Module 1) of the released Boltz-2 affinity checkpoint (`boltz2_aff.ckpt`), exactly reproducing the corresponding hard embedding for one-hot $p_{c,ij}$ at initialization. The embedding E_c enters at the same point as in Boltz-2 through the unchanged `PairwiseConditioning` block.

Equation (1) is the sole interface for explicit geometric conditioning; pocket cropping only selects tokens. We assess the value of the full distribution by comparing this probability-weighted bin embedding with argmax and mean-distance interpolation, and with no explicit distogram input (Section 4.2).

3.3 Affinity Adaptation

Objective. The affinity module outputs a regression value in $\log_{10}$ units and a binary logit for binder-versus-decoy classification. We keep the three loss terms of Boltz-2: a Huber loss on within-assay pairwise differences of predicted and measured affinities, a censoring-aware Huber loss on absolute values, and a focal binary cross-entropy for binder-versus-decoy discrimination. Each training

batch is drawn from a single assay, defined as a ChEMBL assay or a BindingDB Entry DOI. Loss weights, Huber thresholds, and the sampling schedule over data sources are listed in Appendix C.1.

Fine-tuned parameters. The trunk remains frozen. We update all affinity-module parameters, including the input projection, geometry conditioning, eight-layer Pairformer, pooling, output MLPs, and both prediction heads. Both ensemble members are initialized from the eight-layer member (Module 1) of the released Boltz-2 affinity checkpoint.

Homogeneous two-member ensemble. Boltz-2 averages predictions from affinity modules with eight and four Pairformer layers. PairSense instead uses two identically initialized copies of Module 1, allowing ensemble size to vary without changing the member architecture. The members process the same cached inputs in parallel with separate losses, which are averaged without cross-member gradients; predictions are averaged at inference. Ensemble architecture details and training parameters are given in Appendices B.4 and C.1.

3.4 Native-pathway transfer and molecular feature fusion

Native classification pathway. After the Pairformer, each member averages its pair representation over off-diagonal token pairs involving at least one ligand token to obtain a single vector $g_{\mathrm{pre}} \in \mathbb{R}^{128}$. This vector is mapped by the affinity output MLP to $g_{\mathrm{post}} \in \mathbb{R}^{384}$, from which the score MLP and a final linear layer produce the binary logit used for binder-versus-decoy classification during affinity training:

$$g_{\mathrm{pre}} \xrightarrow{\texttt{affinity_out_mlp}} g_{\mathrm{post}} \xrightarrow{\text{score MLP}} \text{binary logit.} \tag{2}$$

We reuse this native binary pathway for CYP substrate prediction.

Transfer to CYP substrate prediction. For the three CYP substrate tasks we do not add a task-specific head or a task identifier. Each training example is a complex of a CYP enzyme with a candidate drug, processed exactly as in Section 3.1, and the binary logit of Equation (2) is trained with a binary cross-entropy against the substrate label. The trunk, the Pairformer, and the pooling are frozen at the affinity-adapted weights; only the affinity output MLP and the layers after it are updated. In PairSense, the input protein supplies enzyme context, so no additional task identifier is used. Examples from the three CYP tasks are pooled for joint training with shared parameters. Each ensemble member is trained with its own loss, and the two predicted probabilities are averaged at inference.

Molecular feature fusion. Molecular fingerprints provide complementary ligand features to the pooled protein–ligand representation. We incorporate these features at the entry of the native binary pathway. With $\phi \in \{0,1\}^{2048}$ the 2048-bit ECFP4 fingerprint (Rogers & Hahn, 2010) of the ligand, $W_\phi \in \mathbb{R}^{128\times 2048}$, and $b_\phi \in \mathbb{R}^{128}$ randomly initialized,

$$g_{\mathrm{pre}} \;\leftarrow\; g_{\mathrm{pre}} + W_\phi\,\phi + b_\phi, \tag{3}$$

which we refer to as `add@gpre`. Only W_ϕ and b_ϕ are new; the rest of the pathway is unchanged. Alternative fingerprint sizes, physicochemical descriptors, concatenation instead of addition, and injection at g_{post} form a search grid. Its search space and configuration recipes are described in Appendix C.4. The native readout and feature-fusion implementation are detailed in Appendix B.5.

4 Binding Affinity and Geometric Representations

We evaluate the use of co-folding representations for binding-affinity prediction and examine the role of their explicit geometric inputs. We first compare the complete PairSense model with Boltz-2 and TerraBind, and then compare alternative representations of the predicted pairwise distance distribution. Figure 1 summarizes the model architecture; data curation and validation-overlap filtering are detailed in Appendices A.1–A.2, benchmark conventions in Appendix A.3, and training and checkpoint selection in Appendices C.1–C.2.

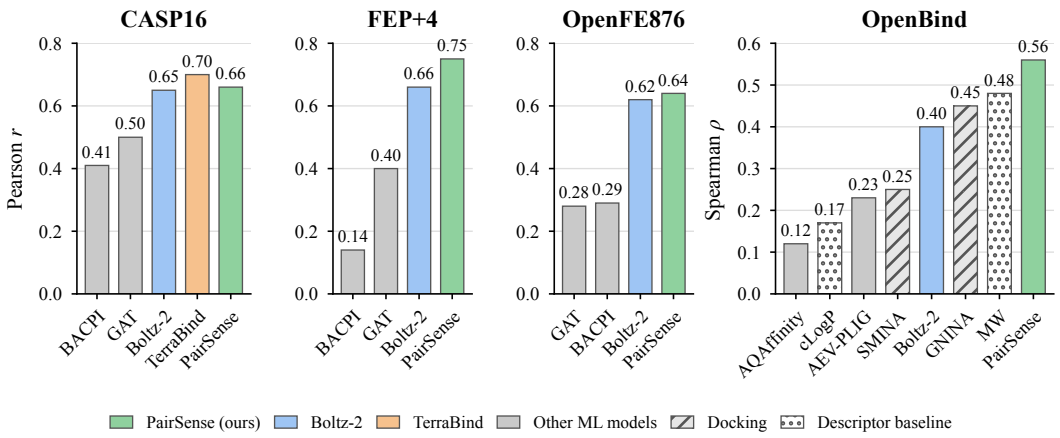

Figure 2: Binding affinity performance of PairSense and baselines on four benchmarks. CASP16, FEP+4 and OpenFE876 report Pearson's r with the molecular-weight correction of Boltz-2; OpenBind reports Spearman's ρ without correction. Baseline sources are given in Section 4.1.

4.1 Binding affinity performance

Baselines. On CASP16 (Gilson et al., 2026), FEP+4 (Ross et al., 2023) and OpenFE876 (OpenFreeEnergy, 2024), we compare against Boltz-2 and the two machine-learning baselines evaluated in its original paper: BACPI (Li et al., 2022), which combines a protein-sequence CNN with a ligand graph attention network, and GAT, its ligand-only variant without the protein branch. Scores for these three methods are taken from Tables 11, 12 and 14 of Passaro et al. (2025). We additionally include TerraBind (Rossi et al., 2026) on CASP16. For OpenBind, we use the baselines released with the benchmark (OpenBind Consortium, 2026), evaluated on the 494 compounds in its first release (May 2026). These comprise Boltz-2, the machine-learning affinity models AEV-PLIG (Valsson et al., 2025) and AQAffinity (SandboxAQ, 2026), the docking methods GNINA (McNutt et al., 2021) and SMINA (Koes et al., 2013), and two single-descriptor rankings based on molecular weight and cLogP.

PairSense achieves higher correlations than Boltz-2 on all four affinity benchmarks (Figure 2). Across all baselines, PairSense achieves the highest correlation on FEP+4, OpenFE876 and OpenBind, and ranks second on CASP16 after TerraBind.[1] The largest gains are observed on FEP+4, where Pearson's r increases from 0.66 to 0.75, and OpenBind, where Spearman's ρ increases from 0.397 to 0.56. Smaller gains are observed on CASP16 and OpenFE876.

Having evaluated the complete model, we next examine its geometric input. Under matched experimental conditions, we compare PairSense's probability-weighted bin embedding with point-estimate representations and with omitting explicit geometry.

4.2 Geometric input representations

The implementation details for the compared geometric inputs are given in Appendix B.3.

We compare four distogram inputs: the full predicted distance distribution, argmax one-hot encoding, mean-distance interpolation, and no explicit distogram. Mean-distance interpolation encodes the expected distance by linearly interpolating between adjacent distance-bin embeddings in the same embedding matrix. The no-explicit-distogram variant receives an all-zero distogram input while retaining all other model inputs. Each input configuration is evaluated with an independently trained model. Checkpoints are selected independently using validation Pearson's r.

[1] TerraBind is not publicly available for independent evaluation; we therefore derive its CASP16 score from the per-target results reported in the original paper. TerraBind reports Pearson's $r = 0.45$ on L1000 (17 ligands) and $r = 0.73$ on L3000 (123 ligands). Following the compound-count-weighted aggregation used by Boltz-2 (Passaro et al., 2025), we obtain $(17 \times 0.45 + 123 \times 0.73)/140 \approx 0.70$. The same aggregation is used for PairSense.

Table 1: Effect of distogram representation on binding affinity prediction. Each variant is trained separately, and checkpoints are selected using the same validation criterion. Results are reported for a single training seed. Bold indicates the highest correlation in each column, including ties at the displayed precision.

Geometric input	CASP16	FEP+4	OpenFE876	OpenBind
Full distribution	**0.66**	0.75	**0.64**	**0.56**
Argmax bin	**0.66**	0.72	0.63	0.54
Mean-distance interpolation	0.62	**0.76**	0.63	0.52
No explicit distogram	0.61	0.73	0.61	0.47

Table 1 distinguishes three explicit geometric representations from the no-explicit-distogram control. Explicit inputs yield higher correlations in 11 of 12 comparisons with this control, the exception being argmax on FEP+4 (0.72 versus 0.73). The full distribution improves over the control by 0.05 on CASP16, 0.02 on FEP+4, 0.03 on OpenFE876, and 0.09 on OpenBind.

Performance also depends on the representation: mean-distance interpolation leads on FEP+4 (0.76 versus 0.75), whereas the full distribution achieves the highest or joint-highest correlation on the other three benchmarks. These single-seed results suggest that explicit geometric conditioning is useful and favor the probability-weighted bin embedding overall.

5 CYP Substrate Prediction

CYP substrate recognition provides a complementary test of protein-conditioned molecular property prediction at an enzyme-dependent metabolic endpoint. We evaluate CYP2C9, CYP2D6, and CYP3A4 using the affinity-adapted interaction representation and native binary readout, rather than treating predicted affinity as a proxy for substrate status. We first evaluate the complete predictor and then examine the contributions of additional affinity adaptation, protein information, joint CYP training, and molecular fingerprints.

5.1 Protein-conditioned CYP substrate prediction

The three CYP tasks provide a complementary setting for testing whether shared supervision can support substrate prediction across enzyme contexts. We therefore use one shared predictor for the three tasks, with each example pairing a molecule with its corresponding CYP protein. During CYP adaptation, the pooled interaction representation g_{pre} is kept fixed, while the native binary prediction pathway is fine-tuned using substrate labels (Figure 1). Enzyme context is provided through the protein input rather than enzyme-specific output heads, so each prediction remains conditioned on the corresponding protein. In the main configuration, a projected 2048-bit molecular fingerprint (Rogers & Hahn, 2010) is added to g_{pre} before the readout MLP. Detailed training and model-selection procedures are provided in Appendix C.3.

Under this protocol, jointly adapted PairSense achieves AUPRCs of 0.536 and 0.763 on CYP2C9 and CYP2D6, respectively, and an AUROC of 0.717 on CYP3A4 (Table 2). It outperforms the jointly trained MiniMol baseline, which receives both CYP task identity and FP2048, on all three tasks. PairSense also exceeds the FP2048-only control, showing that the complete pipeline provides predictive value beyond the molecular fingerprint alone. These comparisons establish the performance of the complete model; we next examine the components underlying these results.

5.2 Contributions of affinity adaptation, protein information, and joint training

The control implementations for Tables 2 and 3 are described in Appendix D.1; their seed-level results are reported in Appendix D.2.

Table 2: CYP substrate prediction under matched supervision. Bold marks the best score in each task. All rows other than the published results use the same five official TDC seeds, are trained jointly on the three CYP tasks, and receive 2048-bit Morgan fingerprints. Controls reuse the recipe of the corresponding PairSense configuration without separate hyperparameter tuning; entries are the mean over the five seeds with the population standard deviation. † Published TDC leaderboard results, taken from a snapshot frozen on 17 September 2026. The three task scores come from different models: MiniMol, ContextPred and CFA, respectively. ‡ Frozen MiniMol embeddings with the downstream MLP head of Klaser et al. (2024). For joint training, MiniMol and FP2048-only use additive conditioning on a three-dimensional task one-hot vector (Appendix D.1).

Model (FP2048-injected)	CYP2C9 Sub. (AUPRC)	CYP2D6 Sub. (AUPRC)	CYP3A4 Sub. (AUROC)
PairSense (joint)	**0.536 ± 0.029**	**0.763 ± 0.029**	**0.717 ± 0.021**
MiniMol‡	0.471 ± 0.045	0.710 ± 0.041	0.701 ± 0.028
FP2048 only	0.406 ± 0.024	0.689 ± 0.022	0.691 ± 0.017
TDC leaderboard SOTA†	0.474 ± 0.025	0.736 ± 0.024	0.667 ± 0.019

Table 3: Decomposition of CYP substrate transfer. Bold marks the best score in each task within each panel. All rows use the same five official TDC seeds; PairSense and the ligand-only-trunk control are read out through the native classification pathway, whereas MiniMol uses the downstream MLP head of Klaser et al. (2024). Entries are the mean over the five seeds with the population standard deviation. (a) Representations compared under joint three-task training with 2048-bit Morgan fingerprints. The pre-fine-tuning row is the actual pretrained initialization used for affinity fine-tuning, not a randomly initialized model. The ligand-only-trunk control excludes protein tokens before the Pairformer. This control and MiniMol receive a one-hot task identity, since their inputs do not identify the enzyme. (b) Joint versus single-task adaptation, with and without fingerprint injection.

Representation			CYP2C9 Sub. (AUPRC)	CYP2D6 Sub. (AUPRC)	CYP3A4 Sub. (AUROC)
(a) Joint training with FP2048					
PairSense, after affinity fine-tuning			**0.536 ± 0.029**	**0.763 ± 0.029**	**0.717 ± 0.021**
PairSense, before affinity fine-tuning			0.481 ± 0.030	0.587 ± 0.050	0.681 ± 0.025
Ligand-only trunk			0.436 ± 0.036	0.644 ± 0.032	0.647 ± 0.019
MiniMol			0.471 ± 0.045	0.710 ± 0.041	0.701 ± 0.028
(b) Training configuration					
Model	**Joint**	**FP2048**			
PairSense	✓	✓	**0.536 ± 0.029**	**0.763 ± 0.029**	**0.717 ± 0.021**
PairSense	✓	–	0.513 ± 0.052	0.708 ± 0.020	0.680 ± 0.030
PairSense	–	✓	0.334 ± 0.043	0.657 ± 0.032	0.612 ± 0.023
PairSense	–	–	0.464 ± 0.029	0.666 ± 0.033	0.598 ± 0.051
MiniMol	✓	✓	0.471 ± 0.045	0.710 ± 0.041	0.701 ± 0.028
MiniMol	✓	–	0.450 ± 0.028	0.667 ± 0.021	0.663 ± 0.036
MiniMol	–	✓	0.393 ± 0.031	0.644 ± 0.040	0.623 ± 0.038
MiniMol	–	–	0.424 ± 0.030	0.561 ± 0.025	0.609 ± 0.064

Additional affinity adaptation. We compare CYP adaptation from the actual initialization used before PairSense affinity training with adaptation from the resulting affinity-trained checkpoint, using the same joint-training and FP2048 configuration (Table 3a). Additional affinity adaptation increases the mean scores from 0.481 to 0.536 on CYP2C9, from 0.587 to 0.763 on CYP2D6, and from 0.681 to 0.717 on CYP3A4. These improvements indicate that the additional binding supervision provides a stronger starting point for transferring the interaction representation and native readout to CYP prediction.

Value of protein information. PairSense's co-folding backbone jointly represents the protein and ligand, providing a natural way to incorporate the corresponding CYP enzyme into substrate prediction. To assess the value of this information beyond molecular inputs and task identity, we construct a ligand-only-trunk control adapted from the placeholder-input construction of Jang et al. (2026, Appendix E), masking the placeholder token and its atoms before input embedding. The ligand-only-trunk control receives CYP task identity during downstream adaptation; implementation details are provided in Appendix D.1.

Under joint training with FP2048, protein-conditioned PairSense outperforms the ligand-only-trunk control by 0.100, 0.119, and 0.070 on CYP2C9, CYP2D6, and CYP3A4, respectively (Table 3a). These results demonstrate the predictive value of incorporating protein information through co-folding representations for joint CYP substrate prediction, beyond molecular features and task identity.

Joint CYP training. Joint adaptation improves PairSense over independent single-task adaptation on all three tasks, both with and without FP2048 (Table 3b). Without fingerprints, the mean gains are 0.049 on CYP2C9, 0.042 on CYP2D6, and 0.082 on CYP3A4, showing that the benefit of joint training is already present before molecular feature augmentation. MiniMol likewise benefits from joint training on all three tasks under the original task-wise partitions. Thus, shared CYP supervision is useful for both the protein-conditioned and task-aware ligand-only models in this setting, rather than being a benefit exclusive to PairSense.

Molecular fingerprint augmentation. Within the joint PairSense configuration, adding FP2048 further increases the mean scores by 0.023, 0.055, and 0.037 on CYP2C9, CYP2D6, and CYP3A4, respectively (Table 3b). Its effect is mixed under single-task adaptation, including a reduction on CYP2C9. Fingerprint augmentation therefore provides a useful complementary input in the final joint configuration, rather than a uniform improvement across training settings.

Together, these comparisons and ablation studies support the use of affinity-adapted co-folding representations and their native readout for CYP substrate prediction. They highlight the benefits of additional affinity adaptation and explicit protein information, while showing how joint training and molecular fingerprint augmentation contribute to the final model's performance under the evaluated protocol.

6 Conclusion

PairSense represents small molecules in their protein context for binding affinity and CYP substrate prediction. It combines frozen co-folding interaction features with full predicted pairwise distance distributions, providing a shared representational basis for these distinct endpoints without generating complex coordinates. On all four affinity benchmarks, PairSense achieves higher correlations than Boltz-2. Controlled geometry comparisons support the use of explicit predicted distance distributions, with advantages over point estimates that vary across benchmarks.

For CYP substrate prediction, additional binding supervision improves performance when the resulting interaction representation is kept fixed and its native binary readout is adapted to substrate labels. The protein-conditioned predictor outperforms the evaluated task-aware ligand-only controls across CYP2C9, CYP2D6, and CYP3A4. Joint CYP training improves over separate task models, while molecular fingerprints provide complementary gains in the joint configuration. Together, these findings support an interaction-based view of molecular property prediction, in which a molecule is represented with its relevant protein and the resulting features can be reused across binding and metabolic endpoints.

Statement on the Use of AI Tools

In this work, we used generative AI tools for research method design, code implementation, data cleaning, translation, and interpretation of experimental results. Experimental designs were discussed with ChatGPT and Claude Code, and every design and setting was decided and approved by the authors before execution. Specifically, AI coding assistants (primarily Claude Code, and to a lesser extent Codex) were used to implement and debug data-processing (including data-cleaning), training-orchestration, and evaluation scripts, and to execute part of the experiments, and to assist in tabulating experimental results; all reported numbers were computed by our own scripts and checked against the stored per-sample predictions. We have not used generative AI tools for hypothesis refinement, and synthetic data generation, theoretical modelling, mathematical propositions, proof assistance, and qualitative analysis are not applicable to this work. Additionally, we used generative AI tools for polishing the writing (ChatGPT), for scientific figure design and layout guidance (ChatGPT; no figures in the paper were directly generated by ChatGPT), and for literature retrieval and analysis (ChatGPT Deep Research). All AI-assisted content was reviewed by the authors: retrieved references and baseline numbers were checked against the original papers, and translated or polished text was checked for faithfulness to the authors' intended claims. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

Ethics Statement

This work uses only publicly available data: binding-affinity measurements from public databases and published benchmarks, and the Therapeutics Data Commons (TDC) ADMET benchmark. No human subjects or personally identifiable information are involved. All ADMET data were obtained through the official TDC data interface and are used as provided, including the original splits, so that our results remain comparable with previously published baselines. Some records in these datasets contain structural inconsistencies that are inherent to the source data; we report them in Appendix A.5 but do not modify them.

Reproducibility Statement

Section 3 specifies the model, the geometric interface and the training objectives. Appendices A.1–A.4 describe the data sources, filtering and splits for both the affinity and ADMET experiments; Appendix B lists architectural details and parameter counts; Appendix C gives optimization settings, including multi-task training and model selection. The PairSense backbone and affinity module are initialized from the publicly released Boltz-2 checkpoints. If the paper is accepted, we will release the training and inference code and scripts in a public GitHub repository with the camera-ready version.

References

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, 2024.

Jonathan B. Baell and Georgina A. Holloway. New Substructure Filters for Removal of Pan Assay Interference Compounds (PAINS) from Screening Libraries and for Their Exclusion in Bioassays. *Journal of Medicinal Chemistry*, 53(7):2719–2740, 2010. doi: 10.1021/jm901137j.

David Buterez, Jon Paul Janet, Steven J. Kiddle, and Pietro Liò. MF-PCBA: Multifidelity High-Throughput Screening Benchmarks for Drug Discovery and Machine Learning. *Journal of Chemical Information and Modeling*, 63(9):2667–2678, 2023. doi: 10.1021/acs.jcim.2c01569.

Miriam Carbon-Mangels and Michael C. Hutter. Selecting Relevant Descriptors for Classification by Bayesian Estimates: A Comparison with Decision Trees and Support Vector Machines Approaches for Disparate Data Sets. *Molecular Informatics*, 30(10):885–895, 2011. doi: 10.1002/minf.201100069.

Chai Discovery, Jacques Boitreaud, Jack Dent, Matthew McPartlon, Joshua Meier, Vinicius Reis, Alex Rogozhnikov, and Kevin Wu. Chai-1: Decoding the molecular interactions of life. *bioRxiv*, 2024. doi: 10.1101/2024.10.10.615955.

Peter J. A. Cock, Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J. L. de Hoon. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 2009. doi: 10.1093/bioinformatics/btp163.

Jeff Daily. Parasail: SIMD C library for global, semi-global, and local pairwise sequence alignments. *BMC Bioinformatics*, 17(1):81, 2016. doi: 10.1186/s12859-016-0930-z.

Jiaojiao Fang, Yan Tang, Changda Gong, Zejun Huang, Yanjun Feng, Guixia Liu, Yun Tang, and Weihua Li. Prediction of cytochrome P450 substrates using the explainable multitask deep learning models. *Chemical Research in Toxicology*, 37(9):1535–1548, 2024. doi: 10.1021/acs.chemrestox.4c00199.

Michael K. Gilson, Jerome Eberhardt, Peter Škrinjar, Janani Durairaj, Xavier Robin, and Andriy Kryshtafovych. Assessment of Pharmaceutical Protein–Ligand Pose and Affinity Predictions in CASP16. *Proteins: Structure, Function, and Bioinformatics*, 94(1):249–266, 2026. doi: 10.1002/prot.70061.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020.

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W. Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

Hyosoon Jang, Hyunjin Seo, Yunhui Jang, Seonghyun Park, and Sungsoo Ahn. Boltz is a strong baseline for atom-level representation learning. *arXiv preprint arXiv:2602.13249*, 2026. Version 1 of arXiv:2602.13249.

Nan Jiang, Mohammed Quazi, Christina Schweikert, D. Frank Hsu, Tudor Oprea, and Suman Sirimulla. Enhancing ADMET property models performance through combinatorial fusion analysis. *ChemRxiv*, 2023. doi: 10.26434/chemrxiv-2023-dh70x.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E Bolton. PubChem 2023 update. *Nucleic Acids Research*, 51(D1):D1373–D1380, 2023. doi: 10.1093/nar/gkac956.

Kerstin Klaser, Blazej Banaszewski, Samuel Maddrell-Mander, Callum McLean, Luis Müller, Ali Parviz, Shenyang Huang, and Andrew W. Fitzgibbon. MiniMol: A parameter-efficient foundation model for molecular learning. In *ICML 2024 Workshop on Efficient and Accessible Foundation Models for Biological Discovery*, 2024.

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):1893–1904, 2013. doi: 10.1021/ci300604z.

Min Li, Zhangli Lu, Yifan Wu, and YaoHang Li. BACPI: a bi-directional attention neural network for compound–protein interaction and binding affinity prediction. *Bioinformatics*, 38(7):1995–2002, 2022. doi: 10.1093/bioinformatics/btac035.

T. Liu, Y. Lin, X. Wen, R. N. Jorissen, and M. K. Gilson. BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities. *Nucleic Acids Research*, 35 (Database):D198–D201, 2007. doi: 10.1093/nar/gkl999.

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*, 13:43, 2021. doi: 10.1186/s13321-021-00522-2.

OpenBind Consortium. OpenBind structure–affinity data release: Enterovirus A71 (EV-A71) / coxsackievirus A16 (CVA16) 2A protease. Zenodo, https://doi.org/10.5281/zenodo.20026661, 2026.

OpenFreeEnergy. IndustryBenchmarks2024: OpenFE industry benchmarking project. https://github.com/OpenFreeEnergy/IndustryBenchmarks2024, 2024. Membership snapshot dda89daf; data snapshot ff41e5ad.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, David Kwabi-Addo, Dominique Beaini, Tommi Jaakkola, and Regina Barzilay. Boltz-2: Towards accurate and efficient binding affinity prediction. *bioRxiv*, 2025.

Protenix Team, Yuxuan Zhang, Chengyue Gong, Hanyu Zhang, Wenzhi Ma, Zhenyu Liu, Xinshi Chen, Jiaqi Guan, Lan Wang, Yanping Yang, Yu Xia, and Wenzhi Xiao. Protenix-v1: Toward high-accuracy open-source biomolecular structure prediction. *bioRxiv*, 2026.

RDKit. RDKit: Open-source cheminformatics. https://www.rdkit.org, n.d. Official software citation; accessed 24 September 2026.

Sereina Riniker and Gregory A. Landrum. Better Informed Distance Geometry: Using What We Know To Improve Conformation Generation. *Journal of Chemical Information and Modeling*, 55 (12):2562–2574, 2015. doi: 10.1021/acs.jcim.5b00654.

David Rogers and Mathew Hahn. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. doi: 10.1021/ci100050t.

Gregory A. Ross, Chao Lu, Guido Scarabelli, Steven K. Albanese, Evelyne Houang, Robert Abel, Edward D. Harder, and Lingle Wang. The maximal and current accuracy of rigorous protein-ligand binding free energy calculations. *Communications Chemistry*, 6(1):222, 2023. doi: 10.1038/s42004-023-01019-9.

Matteo Rossi, Ryan Pederson, Miles Wang-Henderson, Ben Kaufman, Edward C. Williams, Carl Underkoffler, Owen Lewis Howell, Adrian Layer, Stephan Thaler, Narbe Mardirossian, and John Anthony Parkhill. TerraBind: Fast and accurate binding affinity prediction through coarse structural representations. *arXiv preprint arXiv:2602.07735*, 2026.

SandboxAQ. AQAffinity: Open-source structure-free affinity prediction built on OpenFold3. https://huggingface.co/SandboxAQ/AQAffinity, 2026.

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. doi: 10.1038/nbt.3988.

Ísak Valsson, Matthew T. Warren, Charlotte M. Deane, Aniket Magarkar, Garrett M. Morris, and Philip C. Biggin. Narrowing the gap between machine learning scoring functions and free energy perturbation using augmented data. *Communications Chemistry*, 8:41, 2025. doi: 10.1038/s42004-025-01428-y.

Jeremy Wohlwend, Gabriele Corso, Saro Passaro, Noah Getz, Mateo Reveiz, Ken Leidal, Wojtek Swiderski, Liam Atkinson, Tally Portnoi, Itamar Chinn, Jacob Silterra, Tommi Jaakkola, and Regina Barzilay. Boltz-1: Democratizing biomolecular interaction modeling. *bioRxiv*, 2024. doi: 10.1101/2024.11.19.624167.

Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2019. doi: 10.1021/acs.jcim.9b00237.

Yingjie Yang, Yuxin Zhang, Wenxiang Song, Keyun Zhu, Xinmin Li, Mengyu Tong, Guixia Liu, Weihua Li, and Yun Tang. EviCYP: In silico prediction of cytochrome P450 substrates based on vector quantization and evidential deep learning. *Journal of Chemical Information and Modeling*, 66(7):3892–3903, 2026. doi: 10.1021/acs.jcim.6c00074.

Barbara Zdrazil, Eloy Felix, Fiona Hunter, Emma J Manners, James Blackshaw, Sybilla Corbett, Marleen de Veij, Harris Ioannidis, David Mendez Lopez, Juan F Mosquera, Maria Paula Magarinos, Nicolas Bosc, Ricardo Arcila, Tevfik Kizilören, Anna Gaulton, A Patrícia Bento, Melissa F Adasme, Peter Monecke, Gregory A Landrum, and Andrew R Leach. The ChEMBL Database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods. *Nucleic Acids Research*, 52(D1):D1180–D1192, 2024. doi: 10.1093/nar/gkad1004.

A Datasets and Evaluation Protocols

A.1 Binding-affinity data sources and curation

The regression data combine ChEMBL records with complementary BindingDB records (Zdrazil et al., 2024; Liu et al., 2007). ChEMBL uses release 34 and retains single-protein targets with confidence score 9, binding or functional assays, and standard types $\{\mathrm{IC50}, \mathrm{Ki}, \mathrm{Kd}, \mathrm{EC50}, \mathrm{XC50}, \mathrm{AC50}\}$ reported in nM. Records with a nonempty data-validity comment are removed; variant sequences are taken from the recorded variant sequence when present. BindingDB contributes only records absent from ChEMBL. We use the Entry DOI as the assay identifier and exclude records reporting more than one protein chain. For the remaining records, we retain chain 1 and the original affinity relation qualifier. Because a BindingDB entry may span several proteins, about 7% of the pairs entering the pairwise difference term during training, weighted by the assay sampling probabilities, compare measurements on different proteins.

For both regression sources, molecules are standardized by retaining the largest organic fragment and removing formal charge with RDKit cleanup, largest-fragment, and uncharging operations (RDKit, n.d.). Molecules with more than 50 heavy atoms, molecules matching the RDKit PAINS catalogue (Baell & Holloway, 2010), proteins outside 30–1200 residues, and labels outside $[-6, 6]$ in $\log_{10}(\mu\mathrm{M})$ are excluded. The upper protein-length limit is not applied to the held-out validation assays. We do not apply the ChEMBL Structure Pipeline.

Repeated measurements are grouped by assay identifier, molecule identifier, standard type, and protein sequence, and the affinity value is replaced by the group median. The original relation qualifier is kept only when it is consistent within the group; otherwise it is set to equality. Labels are $y = \log_{10}(\mathrm{nM}) - 3$ in $\log_{10}(\mu\mathrm{M})$ units, so smaller values indicate stronger binding, and qualifiers are encoded as $q \in \{-1, 0, +1\}$ for "$<$", "$=$", and "$>$", respectively.

For the hit-to-lead subset, "$<$" records are discarded, as are "$>$" records below $10\,\mu\mathrm{M}$. An assay is retained when it contains at least 10 records, at least 10 unique values, a unique-to-total ratio of at least 0.2, label population standard deviation of at least 0.25, and mean pairwise molecular Tanimoto similarity of at least 0.25; the 16 validation assays are exempt from these assay-level filters. All retained regression rows are marked as hit-to-lead. Unlike Boltz-2 (Passaro et al., 2025), we do not apply its assay-level structure-quality filter, which retains an assay only when the mean predicted interface pTM (ipTM) of ten randomly sampled binders exceeds 0.75. This filter requires diffusion-based structure generation for every candidate assay and screens the quality of generated coordinates. PairSense never uses generated coordinates: its affinity module operates only on trunk representations and predicted distograms. The filter is therefore not required by our diffusion-free setting.

The binary affinity data are assembled from the PubChem HTS portion of the MF-PCBA collection (Buterez et al., 2023). The collection provides 60 datasets containing single-dose and dose-response screens; assay identifiers are resolved to target protein sequences through PubChem and NCBI records (Kim et al., 2023). We require at least 100 compounds per assay and a hit rate below 0.1. Confirmed actives in the dose-response screen are binders, while inactive single-dose compounds are decoys; decoys are subsampled to approximately a $1{:}9$ binder-to-decoy ratio, and PAINS are removed.

This source contributes $224{,}949$ rows from 53 assays and $222{,}804$ cached complexes. We refer to it as PubChem HTS (MF-PCBA) to distinguish the constructed subset from the broader PubChem HTS category used in the original sampling table.

Before sequence-based filtering, each retained hit-to-lead binder is paired with one synthetic decoy sampled from the hit-to-lead molecule pool. A candidate decoy is accepted when its Tanimoto similarity, computed with a Morgan radius-2, 2048-bit ECFP4 fingerprint (Rogers & Hahn, 2010), is below 0.3 to every binder with the same exact target identifier. Restricting this source to rows with available precomputed features yields $1{,}184{,}635$ binders and $1{,}059{,}103$ decoys before sequence-based filtering, totaling $2{,}243{,}738$ rows.

Table 4 lists the composition of the training snapshot ($3{,}249{,}838$ rows) and the validation set ($3{,}247$ rows). By supervision mask, $2{,}204{,}578$ training rows are binary and $1{,}045{,}260$ are regression examples; the training snapshot contains $3{,}354$ unique protein sequences.

Table 4: Composition of the affinity training snapshot and validation set. *Before filtering* counts rows in the source snapshot before benchmark-overlap filtering (Appendix A.2); ChEMBL and BindingDB are counted jointly there, so only their regression subtotal is given. The synthetic-decoy source also contains the paired hit-to-lead binders. The validation assays are not subject to overlap filtering.

Source	Supervision	Before filtering	Training rows
ChEMBL	Regression	–	508,823
BindingDB	Regression	–	536,437
Regression subtotal		1,181,388	1,045,260
PubChem HTS (MF-PCBA)	Binary	224,949	224,949
Synthetic decoys (with paired binders)	Binary	2,243,738	1,979,629
Total training		3,650,075	3,249,838
Validation (16 assays)	Regression	–	3,247

A.2 Affinity validation split and benchmark-overlap filtering

The validation set contains 16 assays from 27 proteins: 9 ChEMBL assays and 7 BindingDB Entry-DOI assays. Its standard types are 3,032 Ki and 215 Kd measurements, and its qualifiers are 3,116 equalities and 131 right-censored values. The validation source counts are 2,479 BindingDB and 768 ChEMBL rows.

Held-out sequence control. To prevent target leakage, we remove from training every protein that is nearly identical to a protein used for evaluation. The evaluation side is a set of 113 query proteins: the protein chains of FEP+4 and OpenFE876, the two CASP16 targets, the 27 proteins of the validation assays, and the OpenBind target. ADMET and PubChem HTS proteins are not queries, because these datasets are not used for affinity evaluation.

Each training sequence is compared with each query by local Smith–Waterman alignment (BLOSUM62, gap-open penalty 10, gap-extension penalty 1). A training sequence is removed when (i) at least 90% of the aligned residue pairs are identical and (ii) the alignment contains at least $\min(50, \lceil 0.8\, L_{\text{query}} \rceil)$ residue pairs, where L_{query} is the query length. Gap columns are excluded from the identity denominator: many benchmark sequences are engineered crystallization constructs with internal deletions, and counting these gaps would let near-identical proteins fall below the threshold.

The alignments are computed with Parasail (Daily, 2016). The exclusion list additionally keeps every protein flagged by an earlier MMseqs2-based filter (Steinegger & Söding, 2017) and a few borderline proteins whose identity lies at the 0.90 threshold under an independent re-check, so that it is never smaller than either method alone. The two CASP16 targets are also removed by their UniProt accessions. Removal applies to all training rows of an excluded protein, across the regression, PubChem HTS, and synthetic-decoy sources; in total, 207 protein sequences are excluded.

To verify the result with a different implementation, we re-align all 113 queries against the 3,354 remaining training sequences with Biopython's PairwiseAligner (Cock et al., 2009) (379,002 pairs). No remaining training sequence meets the exclusion criterion.

A.3 Affinity benchmarks and evaluation conventions

Affinity benchmark protocol. CASP16 contains 140 complexes for two targets (Gilson et al., 2026): 17 ligands for L1000 and 123 for L3000. We report Pearson correlation, calculating it separately for each target and taking a compound-count-weighted average, following Boltz-2 (Passaro et al., 2025); predictions are reported with the molecular-weight correction.

FEP+4 contains 87 neutral compounds across CDK2, TYK2, JNK1, and P38 (Ross et al., 2023), with 16, 16, 21, and 34 compounds per target, respectively. Its labels are free energies in kcal/mol reconstructed from the original IC50 values. We report target-weighted Pearson correlation using the molecular-weight-corrected predictions for the primary comparison.

OpenFE876 contains 876 complexes grouped into 59 system-group/system-name groups, representing approximately 42 biological proteins and 55 unique receptor sequences (OpenFreeEnergy, 2024). Labels are experimental free energies in kcal/mol, with $\Delta G = RT\ln(C)$. We report a compound-count-weighted Pearson correlation over the 59 groups with molecular-weight-corrected predictions.

OpenBind uses the May release, in which 494 compounds are represented by 649 crystal-structure rows; a compound may appear in several rows, and the rows carry 499 unique SMILES. The target is EV-A71 2A protease represented by the CVA16 sequence used in the benchmark. We predict each unique SMILES once, average the raw outputs v_1 and v_2 of the two members as $\widehat{v} = (v_1 + v_2)/2$, convert it as $\widehat{\mathrm{p}K_D} = 6 - \widehat{v}$, assign them to the 649 rows, and average over the rows of each compound to obtain 494 compound-level predictions. Labels are per-compound means of $\mathrm{p}K_D$ over repeated measurements. We report raw-score Spearman correlation over the 494 compounds; no molecular-weight correction is applied.

For CASP16, FEP+4, and OpenFE876, we use the molecular-weight correction convention from the original Boltz-2 evaluation (Passaro et al., 2025). The coefficients are copied unchanged and are not fitted on any of these benchmarks. The correction is applied only after averaging the two member predictions:

$$\widehat{y} = 1.03525938\,\frac{v_1 + v_2}{2} - 0.59992683\,\mathrm{MW}^{0.3} + 2.83288489. \tag{4}$$

The stored predictions remain raw values, and OpenBind is evaluated only in that raw convention.

A.4 CYP DATASETS, PROTEIN INPUTS, AND OFFICIAL SPLITS

The CYP data comprise three substrate-recognition tasks from the Carbon–Mangels collection (Carbon-Mangels & Hutter, 2011): CYP2C9, CYP2D6, and CYP3A4. Their metrics are AUPRC, AUPRC, and AUROC, respectively. The tasks contain 669, 667, and 670 rows; each official test split has 135 rows, leaving train–validation sets of 534, 532, and 535 rows.

For each of the five official seeds, the task-specific training and validation counts are $467/67$ for CYP2C9, $465/67$ for CYP2D6, and $468/67$ for CYP3A4. The positive counts in the test splits are $38/135$, $43/135$, and $79/135$; the corresponding train–validation positive counts are $103/534$, $148/532$, and $276/535$.

The official TDC scaffold partitions (Huang et al., 2021) are constructed separately for each task and used unchanged for both single-task and joint training. All three tasks use the original SMILES and the five official seeds.

The protein input is fixed within each task. The canonical CYP2C9, CYP2D6, and CYP3A4 sequences contain 490, 497, and 503 residues, respectively, and their existing multiple-sequence alignments are used for featurization. The corresponding UniProt accessions are P11712, P10635, and P08684.

A.5 DATA-QUALITY AUDIT AND PREPROCESSING COVERAGE

We initially considered eight protein-related tasks from the TDC ADMET benchmark (Huang et al., 2021): substrate recognition and inhibition for each of CYP2C9, CYP2D6, and CYP3A4, together with P-glycoprotein inhibition and hERG channel blockade. Applying Boltz-2 to these tasks requires preparing a three-dimensional reference conformer for each ligand to construct the input features, including S_{in}. This ligand-input preparation uses RDKit ETKDG (Riniker & Landrum, 2015) and precedes the co-folding trunk; it is separate from diffusion-based generation of a protein–ligand complex, which PairSense does not run.

In the original preprocessing workflow, 53 test molecules could not be processed: 11 were rejected by the upstream metal filter and 42 failed ETKDG conformer generation. These failures occurred in the three CYP inhibition tasks and the P-glycoprotein task. Although ETKDG is stochastic, repeated attempts did not reliably resolve the difficult cases. We examined fallback procedures and recovered inputs for these test molecules, but some required relaxing stereochemical constraints or substituting planar coordinates, which can alter or remove the stereochemical information carried by the reference geometry. Under the preprocessing workflow without these fallbacks, each of the

five excluded tasks has missing inputs in the training–validation set or the test set. We therefore restrict the reported experiments to the three CYP substrate tasks, whose training, validation, and test inputs are complete without these fallbacks. Their common substrate-recognition setting also supports joint training.

Separately, duplicate SMILES and conflicting labels prompted us to check whether the recorded molecular structures matched the compound identities. Across the eight tasks, the original test tables contain six pairs of records with identical SMILES, four of which have conflicting labels. Within the three substrate tasks, CYP2D6 has one conflicting pair, CYP3A4 has one pair with matching labels, and CYP2C9 has none. We compared the names or identifiers in the `Drug_ID` field with structure records in PubChem and ChEMBL (Kim et al., 2023; Zdrazil et al., 2024). This check revealed cases in which distinct compounds had been assigned the same SMILES. For example, bupivacaine, a racemic mixture, and levobupivacaine, its single S-enantiomer, share the same recorded SMILES in the two affected CYP substrate datasets. Such cases concern compound-to-structure correspondence, rather than necessarily indicating contradictory measurements of the same compound.

These findings motivated a separate table of candidate corrections based on PubChem structures. We do not use those replacements in the reported experiments: changing the recorded structures, particularly duplicate or conflicting test records, would change the evaluation inputs relative to the official benchmark used for published comparisons. We retain the original TDC SMILES, labels, and splits, and disclose the identified inconsistencies as limitations of the benchmark rather than silently correcting them.

B Model Architecture and Implementation

B.1 Cached representations and pocket cropping

For a tokenized protein–ligand complex with L full-length tokens, the input embedder produces the single-token representation $S_{\text{in}} \in \mathbb{R}^{L\times 384}$. In this implementation S_{in} is the input-embedder output `s_inputs`, rather than a post-trunk single-token representation. The frozen trunk also produces a pair representation $Z \in \mathbb{R}^{L\times L\times 128}$ and distogram logits $D \in \mathbb{R}^{L\times L\times 64}$.

For the affinity geometry interface, the distance-bin boundaries are 63 evenly spaced values from 2 to 22 Å, which define 64 bins. The two open-ended bins use centers 1.5 Å and 24.5 Å; the other centers d_b are the midpoints of adjacent boundaries.

For pocket selection, a softmax is applied to D and the expected distance for each token pair is computed as

$$\mathbb{E}[d_{ij}] = \sum_b p_{ijb} d_b. \tag{5}$$

For every protein token, the minimum expected distance to a ligand token is taken. Protein tokens with this minimum below 15 Å are retained, and all ligand tokens are retained. Non-protein tokens that are not affinity ligand tokens are removed. The full-length trunk is therefore evaluated before this pocket crop. If the retained set has N tokens, the cropped $(S_{\text{in},c}, Z_c, D_c)$ values are cached for subsequent stages without a token-count cap in the dump path; the cache stores logits D_c, and later computes $p_c = \mathrm{softmax}(D_c)$ when needed.

For comparison, the official Boltz-2 pipeline uses diffusion-generated coordinates to define a crop of at most 256 tokens, reruns the trunk on that crop, generates five diffusion samples, and selects the sample with the highest ipTM to construct the hard distogram for affinity prediction. Each hard distance is assigned to a bin by counting the boundaries strictly below it.

B.2 Affinity module, pair masking, and pooling

Each affinity member combines linear projections of the cropped single-token representation $S_{\text{in},c}$ and the normalized pair representation Z_c to form its pair input. The cropped distance distributions are mapped to 128-dimensional features by a bias-free linear projection initialized from the pretrained distance-bin embeddings, and incorporated through the original pairwise conditioning block. An eight-layer pair-only Pairformer then updates ligand–protein, protein–ligand, and ligand–ligand pairs, with protein–protein pairs masked out. After these updates, masked mean pooling over

the retained pairs, excluding the diagonal, produces $g_{\mathrm{pre}} \in \mathbb{R}^{128}$. An output MLP maps this vector to $g_{\mathrm{post}} \in \mathbb{R}^{384}$, which feeds separate regression and binary-classification branches. Apart from the geometric-input interface and ensemble construction, the module follows the Boltz-2 architecture.

B.3 Geometric-input implementations

All four variants use the same cached trunk outputs and the distance-bin definitions in Appendix B.1, differing only in the explicit geometric input supplied to the affinity module.

The full-distribution variant applies the bias-free soft projection to the softmax distribution $p_{c,ij} = \mathrm{softmax}(D_{c,ij})$ from the cached cropped logits. The argmax variant takes the largest distogram logit and sends its one-hot vector through the same projection; taking the argmax before or after softmax is equivalent. The mean-interpolation variant computes $\mu = \mathbb{E}[d_{ij}] = \sum_b p_{c,ijb} d_b$, locates the adjacent centers $d_k \le \mu < d_{k+1}$, and uses

$$E_{c,ij} = (1-\alpha)e_k + \alpha e_{k+1}, \qquad \alpha = \frac{\mu - d_k}{d_{k+1} - d_k}. \tag{6}$$

The interpolation coefficient is clipped to $[0, 1]$. The outer interpolation intervals span 0.661 Å and 2.661 Å, respectively, so distances above 21.839 Å are represented by interpolation between the final two vectors. The index is clamped to the final adjacent pair when μ reaches the right endpoint, so the endpoint uses e_{63} and e_{64} with $\alpha = 1$ under the one-based bin notation.

The no-explicit-geometry variant sets the projected distance feature to zero, while the pairwise conditioner continues to run. All other cached-input processing is unchanged.

B.4 Ensemble construction and parameter accounting

The two members are constructed by copying the complete eight-layer member state into the second member after replacing its pair-stack depth with eight layers. The original four-layer member weights are discarded, and the two members have bit-identical parameters at initialization. Including the soft projection, one eight-layer member has $5{,}498{,}884$ parameters and the two-member model has $10{,}997{,}768$ parameters. These counts also include the $8{,}192$-parameter hard embedding retained in each member's state; the soft forward path does not use this embedding, so it receives no gradient.

Each member computes its complete loss independently, and the member losses are averaged. At inference, the two predictions are averaged. Gradient clipping is applied to the total model gradient norm, so this operation is a shared scaling step across the members.

B.5 Native CYP readout and molecular feature fusion

During CYP adaptation, the input is a cached g_{pre} vector of dimension 128 for each member. The trunk, Pairformer, and pooling are not recomputed. The native classification path loads `affinity_out_mlp`, the score MLP, and the final binary-logit projection from the affinity module; the regression head is not instantiated. The transferred path is therefore

$$\begin{aligned} g_{\mathrm{pre}} &\xrightarrow{\texttt{affinity_out_mlp}} g_{\mathrm{post}}, \\ g_{\mathrm{post}} &\xrightarrow{\texttt{to_affinity_pred_score}} \text{score} \xrightarrow{\texttt{to_affinity_logits_binary}} \text{logit}. \end{aligned} \tag{7}$$

For fingerprint augmentation, a 2048-bit Morgan fingerprint with radius 2 is projected by a separately initialized affine map $W_\phi \phi + b_\phi$ from 2048 to 128 channels and added to g_{pre}. Binary fingerprint columns are not z-score standardized; any continuous feature standardization is fit on the training rows of each seed.

The two CYP members use binary cross-entropy with logits, average their losses, and average their predicted probabilities at inference.

C Training and Model Selection

C.1 Affinity optimization and assay sampling

The affinity stage trains both eight-layer members while keeping the trunk frozen. We use AdamW with a fixed learning rate of 10^{-4}, weight decay 10^{-3}, and gradient-norm clipping at 10, without a learning-rate scheduler or warmup. Training uses FP32 arithmetic, activation checkpointing, a dropout probability of 0.25, and a global seed of 0; early stopping and exponential moving averages are disabled. AdamW uses $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$, with a single parameter group applying weight decay without bias or LayerNorm exemptions.

Our training data cover the three source categories with the largest sampling weights in the Boltz-2 training mixture: ChEMBL/BindingDB affinity measurements, PubChem HTS data, and synthetic decoys derived from ChEMBL/BindingDB. Together, these categories account for 94% of the sampling probability in Boltz-2 (Table 8 of Passaro et al., 2025). Our PubChem HTS data are drawn from the MF-PCBA collection rather than the full HTS collection used by Boltz-2. We omit the remaining categories—PubChem small assays with continuous or binary labels, CeMM fragments, and MIDAS metabolites—and retain the original relative weights of 0.25, 0.44, and 0.25 for the three included categories. Renormalizing these weights gives source-selection probabilities of approximately 26.6%, 46.8%, and 26.6%, respectively. These probabilities govern the choice of source for each assay batch, independently of the number of records in each source.

For regression, an assay is selected with a weight of IQR+0.01, and five examples are sampled without replacement; if fewer than five are available, all available examples are used. For binary data, one binder is sampled first, followed by four non-binders from the same assay. The non-binders are sampled without replacement when at least four are available, and with replacement otherwise, giving a batch of five examples with a one-to-four positive-to-negative ratio.

The affinity objective is

$$\mathcal{L} = 0.9\,\mathcal{L}_{\text{dif}} + 0.1\,\mathcal{L}_{\text{abs}} + 1.0\,\mathcal{L}_{\text{binary}}. \tag{8}$$

Both regression terms use a Huber threshold of 0.5. The difference term is computed only for uncensored examples from the same assay, over unordered pairs $i < j$. The absolute term is censoring-aware: a right-censored value is penalized only when the prediction is below its bound, and a left-censored value only when the prediction is above its bound. The binary term is focal binary cross-entropy with focusing exponent $\gamma = 1.0$; for each member, the positive and negative class weights are λ and $1 - \lambda$, with $\lambda = 0.8$ for Module 1 and $\lambda = 0.6$ for Module 2.

Training uses distributed data parallelism on eight NVIDIA A100 GPUs, with one independently sampled assay batch per GPU and no gradient accumulation; unused parameters are permitted in the DDP configuration. Each optimizer update therefore aggregates up to 40 examples from eight assay draws, which may select the same assay more than once. Each epoch comprises 500 optimizer updates, corresponding to 4000 assay batches across all devices. A 50-epoch run takes approximately 12–13 hours, or about 15 minutes per epoch.

C.2 Affinity validation and checkpoint selection

The reported affinity point uses the same pair of member checkpoints for all four benchmarks, with the benchmark aggregation protocols defined in Appendix A.3.

At the end of each training epoch, we evaluate the model on the validation set comprising 3,247 measurements from the 16 assays specified by Boltz-2. For each example, we average the regression predictions of the two ensemble members from the same epoch, using raw predictions in $\log_{10}(\mu\text{M})$ without molecular-weight correction. We compute Pearson's correlation coefficient separately within each assay and take the unweighted mean across assays as the validation score. Censored measurements are included using their recorded threshold values. We save the checkpoint achieving the highest validation score during training and use it for reporting.

C.3 CYP OPTIMIZATION AND CHECKPOINT SELECTION

CYP adaptation uses `torch.optim.Adam`, with weight decay in Adam's coupled L2 form rather than AdamW. Two parameter groups are used: the output MLP group receives the configured learning-rate multiplier, while the remaining trainable parameters use the base learning rate.

The reported joint configuration trains the output MLP and subsequent classification layers, adds the 2048-bit fingerprint at g_{pre}, trains for 200 epochs with batch size 512, uses binary cross-entropy, a constant learning- rate schedule without warmup, and disables positive-class weighting and early stopping. The reported learning rate is 2×10^{-3}, the weight decay is 3×10^{-3}, and the output-MLP multiplier is 1.0.

Joint training and model selection. Joint training mixes examples from the three task-specific scaffold partitions through one native binary pathway. PairSense has no task-specific output head: the CYP protein in the input supplies task context. Controls without protein input encode the CYP task as a three-dimensional one-hot vector, which is projected through a trainable affine map and added to the input representation of the downstream MLP. For the ligand-only-trunk control, each ensemble member projects this vector to 128 dimensions and adds it to g_{pre} before the output MLP. MiniMol and FP2048-only apply the same additive task-conditioning principle to their 512-dimensional embedding and 2048-dimensional fingerprint, respectively; implementation details are given in Appendix D.1. For each seed, the selected epoch is the highest validation score for that task; under joint training, the three official metrics are averaged with equal weight and the same epoch is used for all tasks. We report the mean $\pm$ population standard deviation over the five official seeds at the selected epoch.

At each epoch, the training rows are randomly permuted once and then split into batches. Training is CPU-only with one thread per process; each seed uses its own random generator in addition to the global seed.

For ablation rows without their own protein-conditioned representation, the control uses the recipe copied from the corresponding PairSense cell rather than a separately tuned recipe.

C.4 HYPERPARAMETER SEARCH SPACES AND CONFIGURATION-SPECIFIC RECIPES

The native-head grid contains $232 = 29 \times 2 \times 4$ configurations: 29 injection choices, two trainable ranges, and four task groups (three single-task groups and one joint group). The two trainable ranges are the prediction-score head and the output-MLP head. The prediction-score range updates the score MLP and its binary projection; the output-MLP range also unfreezes the output MLP. The injection choices are no injection, plus addition or concatenation at g_{pre} or g_{post} for seven feature sets: descriptors, FP128, FP1024, FP2048, and each of the three fingerprint sizes concatenated with descriptors. The fingerprint features are binary ECFP4 features (Rogers & Hahn, 2010) (Morgan radius 2) without standardization. The descriptor vector has 26 dimensions, computed with RDKit from the original TDC SMILES and z-scored on the training rows. Twelve dimensions describe charge: net formal charge, the numbers of positively and negatively charged atoms, the number of fragments, the formal charge of the largest fragment, a flag indicating whether Gasteiger charges are defined for all atoms, the maximum, minimum, and summed absolute Gasteiger charges (zero when undefined), and the counts of basic amines, acidic groups (carboxylic acid, tetrazole, and N–H sulfonamide), and quaternary nitrogens. Fourteen dimensions are physicochemical: molecular weight, Crippen logP, topological polar surface area, hydrogen-bond donors and acceptors, rotatable bonds, rings, aromatic rings, saturated rings, fraction of sp^3 carbons, heavy atoms, heteroatoms, Crippen molar refractivity, and QED.

The grid uses a ten-value learning-rate pilot, followed by combinations of four weight-decay settings, three output-MLP learning-rate multipliers, two positive-class weighting settings, and two schedules, including cosine decay. For this 232-configuration grid, the batch size is 512.

The grid and the main table differ in both selection protocol and recipe coverage. The grid includes both trainable ranges and all descriptor/fingerprint choices, and selects each configuration using its validation search. The main table reports the specified FP2048 configurations and uses the corresponding PairSense recipe for controls to facilitate ablation comparisons under matched training hyperparameters, without separate tuning of the controls.

Table 5 lists the recipe of each PairSense configuration in Table 3b. All configurations train the output-MLP range for 200 epochs with batch size 512, binary cross-entropy, a constant learning-rate schedule, and no early stopping. When positive-class weighting is enabled, the weight is the ratio of negative to positive examples in the training rows of the run. Each control uses the recipe of the matching row; the output-MLP multiplier does not apply to MiniMol.

Table 5: Per-configuration recipes for the PairSense rows of Table 3b. Single-task configurations use a separate recipe for each task.

Joint	FP2048	Task	LR	Weight decay	MLP mult.	Pos. weight
✓	✓	all three	2×10^{-3}	3×10^{-3}	1.0	off
✓	–	all three	1×10^{-2}	1×10^{-4}	0.3	on
–	✓	CYP2C9	3×10^{-2}	1×10^{-2}	1.0	on
–	✓	CYP2D6	3×10^{-3}	1×10^{-2}	1.0	off
–	✓	CYP3A4	1×10^{-4}	1×10^{-3}	1.0	off
–	–	CYP2C9	3×10^{-2}	0	0.3	on
–	–	CYP2D6	3×10^{-2}	1×10^{-4}	1.0	off
–	–	CYP3A4	3×10^{-3}	1×10^{-2}	1.0	on

D CYP Controls and Additional Results

This appendix gives CYP substrate results and implementation details on the original TDC scaffold splits. CYP2C9 and CYP2D6 are scored by AUPRC, whereas CYP3A4 is scored by AUROC. Unless stated otherwise, every reported aggregate is the mean over the five official seeds with the population standard deviation (ddof=0).

Our ligand-only-trunk control is adapted from the placeholder-input construction of Jang et al. (2026). Their approach uses pooled trunk representations with external probes. Our control instead retains PairSense's affinity module and native prediction head, with the input masking and task conditioning described below.

The three tasks use the original TDC SMILES and the official train/validation/ test partitions. Training mixes examples from the three tasks for joint runs; the native PairSense pathway has no task-specific output head, so the CYP enzyme is identified by the protein input. The ligand-only and molecular-only controls receive a three-dimensional task one-hot vector when they are trained jointly.

D.1 Baseline and control implementations

The ligand-only-trunk control replaces the CYP protein sequence with a single placeholder residue X and uses an empty MSA. Before the input embedder runs, the forward pass sets the placeholder's token pad mask and the pad masks of its associated atoms to zero, treating them as padding rather than valid inputs. This masking is intended to prevent the placeholder from participating in ligand feature computation from the input-embedding stage onward. The same placeholder and ligand block are used for all three tasks, and the cached input is identical across tasks for a given molecule. During joint adaptation, task identity is injected at g_{pre} as `add@gpre:task3`: each member uses a $\text{Linear}(3, 128)$ projection, adding 512 parameters per member.

In the released implementation of Jang et al. (2026), input embedding is computed with the placeholder present, after which its token position is removed by slicing before the Pairformer updates. Our implementation instead retains the placeholder position in the input tensors but masks both its token and atoms before input embedding. The two approaches therefore differ in when and how the placeholder is excluded; our control is an adaptation of their input construction, not a reproduction of their full prediction pipeline.

MiniMol (Klaser et al., 2024) uses a frozen 512-dimensional representation. A z-score transform is fit on training rows only. Its downstream head is a three-layer MLP with hidden width 1024, dropout

0.1, and ReLU. For the FP-add rows, a bias-bearing map $A : 2048 \to 512$ projects FP2048 and adds it to the frozen embedding. For the joint rows, an independent bias-bearing map $B : 3 \to 512$ likewise adds the task one-hot vector. The single-task FP-add row uses A only. The no-FP joint MiniMol row uses B only. We use additive feature injection to preserve each model's existing prediction-head architecture and input dimensionality. Auxiliary features are projected to the dimensionality of the representation entering the head and added to that representation. Accordingly, PairSense receives the projected features at its 128-dimensional g_{pre}, whereas MiniMol receives them at its 512-dimensional molecular embedding. This design incorporates additional information while retaining each model's original head, with only the feature-projection layers added.

The FP2048-only control uses the same downstream MLP shape as MiniMol with a 2048-dimensional fingerprint input. In the joint setting, a trainable affine map $B_{\mathrm{FP}} : 3 \to 2048$ projects the task one-hot vector t and adds it to the fingerprint f, giving $f + B_{\mathrm{FP}}(t)$ as the MLP input. This map includes a bias term and preserves the 2048-dimensional input width.

D.2 Seed-level results

Table 6 gives the seed-level values for all principal configurations; columns are ordered by seed 1 through seed 5. The "before affinity adaptation" row is the exact initialization used for the subsequent affinity adaptation, while the remaining PairSense rows use the affinity-adapted checkpoint.

Table 6: CYP substrate results for all principal configurations. Each row contains the five official TDC seed values in seed order; "joint" mixes all three CYP tasks, "single" trains one CYP task alone, and "no FP" omits the fingerprint. The FP2048-only single-task row is included for completeness and is not a row in the main text. Means and standard deviations reported elsewhere are computed from unrounded seed values and may differ in the last digit from statistics recomputed from this table.

Configuration	S1	S2	S3	S4	S5
(a) CYP2C9 substrate (AUPRC)					
PairSense, joint + FP2048	0.529	0.566	0.496	0.572	0.520
PairSense, joint, no FP	0.585	0.549	0.432	0.486	0.511
PairSense, single + FP2048	0.338	0.320	0.349	0.267	0.398
PairSense, single, no FP	0.483	0.422	0.470	0.440	0.504
Before affinity adaptation, joint + FP2048	0.431	0.475	0.499	0.520	0.478
Ligand-only trunk, joint + FP2048	0.506	0.425	0.433	0.403	0.414
MiniMol, joint, FP add + task add	0.539	0.403	0.447	0.472	0.495
MiniMol, single, FP add	0.388	0.420	0.405	0.418	0.335
MiniMol, joint, no FP, task add	0.441	0.470	0.490	0.408	0.441
MiniMol, single, no FP	0.445	0.377	0.454	0.399	0.447
FP2048 only, joint	0.379	0.450	0.401	0.392	0.408
FP2048 only, single	0.330	0.374	0.326	0.343	0.383
(b) CYP2D6 substrate (AUPRC)					
PairSense, joint + FP2048	0.756	0.806	0.776	0.715	0.762
PairSense, joint, no FP	0.709	0.725	0.702	0.674	0.731
PairSense, single + FP2048	0.653	0.714	0.624	0.633	0.661
PairSense, single, no FP	0.612	0.646	0.683	0.693	0.697
Before affinity adaptation, joint + FP2048	0.534	0.595	0.671	0.536	0.596
Ligand-only trunk, joint + FP2048	0.665	0.683	0.657	0.611	0.602
MiniMol, joint, FP add + task add	0.742	0.750	0.680	0.734	0.644
MiniMol, single, FP add	0.635	0.606	0.645	0.613	0.719
MiniMol, joint, no FP, task add	0.677	0.677	0.675	0.625	0.684
MiniMol, single, no FP	0.577	0.552	0.587	0.516	0.571
FP2048 only, joint	0.703	0.713	0.695	0.684	0.650
FP2048 only, single	0.701	0.668	0.704	0.680	0.671

Table 6 (continued)

Configuration	S1	S2	S3	S4	S5
(c) CYP3A4 substrate (AUROC)					
PairSense, joint + FP2048	0.727	0.684	0.714	0.748	0.713
PairSense, joint, no FP	0.665	0.643	0.706	0.664	0.723
PairSense, single + FP2048	0.620	0.623	0.627	0.565	0.625
PairSense, single, no FP	0.630	0.603	0.515	0.579	0.664
Before affinity adaptation, joint + FP2048	0.702	0.635	0.671	0.697	0.699
Ligand-only trunk, joint + FP2048	0.642	0.644	0.634	0.684	0.632
MiniMol, joint, FP add + task add	0.703	0.669	0.708	0.747	0.677
MiniMol, single, FP add	0.662	0.642	0.656	0.586	0.570
MiniMol, joint, no FP, task add	0.685	0.631	0.704	0.609	0.683
MiniMol, single, no FP	0.665	0.616	0.622	0.486	0.657
FP2048 only, joint	0.681	0.672	0.710	0.678	0.713
FP2048 only, single	0.657	0.641	0.654	0.662	0.639

D.3 Published leaderboard references

A leaderboard snapshot queried on 17 September 2026 reports means and population standard deviations of 0.474 ± 0.025 for CYP2C9 AUPRC (MiniMol (Klaser et al., 2024)), 0.736 ± 0.024 for CYP2D6 AUPRC (ContextPred (Hu et al., 2020)), and 0.667 ± 0.019 for CYP3A4 AUROC (CFA (Jiang et al., 2023)). These are published references from three different models and are not reproductions in our pipeline.