DegradeQuery: Counterfactual Tuple Pretraining for Context-Aware PROTAC Degradation Prediction

Dong Xu[1,3,4], Zhangfan Yang[1,2], Jiantao Wu[1,3], Zexuan Zhu[1,3], Jianqiang Li[1,3], Junkai Ji[1,3,4,†]

[1]School of Artificial Intelligence, Shenzhen University
[2]School of Computer Science, University of Nottingham Ningbo
[3]Intelligent Computing Center, Shenzhen University, 518060 Shenzhen, China
[4]EasternDawn

Abstract

Proteolysis-targeting chimeras (PROTACs) induce protein degradation by recruiting a target protein to an E3 ubiquitin ligase, making degradation a joint outcome of the degrader molecule and its biological context. Although public databases contain thousands of structured molecule–target–E3 records, degradation measurements are available for only a small fraction of them. Existing supervised approaches therefore leave most recorded chemical–biological relationships unused. We introduce DegradeQuery, a context-aware prediction framework that converts these label-missing records into a pretraining signal. Its counterfactual tuple pretraining objective contrasts recorded tuples with alternatives formed by replacing the target, the E3 ligase, or both, enabling the model to learn contextual associations without assigning activity pseudo-labels. The resulting representation is then fine-tuned to predict degradation from the complete molecule–target–E3 context. DegradeQuery outperforms the compared methods on the official PROTAC-8K benchmark, with an area under the ROC curve (AUROC) of 0.9065 and an accuracy of 0.8500. Controlled analyses show that tuple-level pretraining drives the improvement, retains its benefit when using only label-missing records, and complements protein language model representations. These findings demonstrate that incompletely labeled PROTAC databases contain useful relational supervision and provide a practical route for learning context-aware degradation predictors from scarce experimental labels. Code is available at `https://github.com/SZU-ADDG/DegradeQuery`.

1 Introduction

Proteolysis-targeting chimeras (PROTACs) have fundamentally changed how small molecules can modulate protein function. Unlike conventional inhibitors, which typically act by occluding a binding site, PROTACs recruit a target protein into proximity with an E3 ubiquitin ligase, thereby promoting target ubiquitination and subsequent proteasomal degradation [1–4]. This induced-proximity mechanism enables catalytic removal of the target in cells [5] and has been successfully exploited through E3 recruiters such as cereblon ligands [6, 7]. It further broadens the druggable target space to proteins that are difficult to modulate via classical occupancy-driven inhibition [8, 4]. At the same time, this conditional mechanism renders degradation prediction considerably more challenging. A degrader molecule should not be modeled as having a single context-independent degradation label; its measured effect depends jointly on the molecular structure of the degrader, the target protein, the recruited E3 ligase, and the cellular context in which the assay is performed. PROTAC

This work is supported by the Intelligent Computing Center of Shenzhen University.
[†]Corresponding author: Junkai Ji (e-mail: `jijunkai@szu.edu.cn`).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

degradation is therefore not solely a molecular property but rather a context-dependent outcome of the molecule–target–E3 system.

This distinction carries direct implications for data-driven degradation modeling. Public PROTAC resources now provide structured records comprising degrader molecules, target proteins, E3 ligases, warheads, linkers, and assay annotations [9–12]. Related resources for molecular glues similarly catalog chemically induced degradation entries by small molecule, recruited protein, and target context [13, 14]. Yet reliable degradation labels remain scarce. Quantities such as the half-maximal degradation concentration (DC_{50}), maximum degradation ($D_{\max}$), and measured percent degradation are costly to obtain and consequently unavailable for many entries [9, 10]. As a result, a substantial portion of available data contains chemical and biological context but lacks a complete degradation label. Treating PROTAC prediction as a standard supervised molecular classification problem leaves this context underutilized and encourages models to learn primarily from the labeled subset, even though unlabeled entries still describe which molecules were studied in which target–E3 contexts.

Recent computational methods have advanced PROTAC degradation prediction along several directions. DeepPROTACs demonstrated that deep learning can predict targeted degradation from PROTAC-derived representations [15]. PROTAC-Degradation-Predictor extended this line of work by incorporating assay and target context into machine-learning-based prediction [16]. More recent models have moved toward explicit ternary or structure-aware representations: PROTAC-STAN employs a structure-informed deep ternary attention framework [17], and DegradeMaster adopts a semi-supervised E(3)-equivariant graph neural network [18]. Complementary work on ternary complex prediction and biomolecular interaction modeling reinforces this perspective, indicating that the relevant computational object is not an isolated molecule but a multi-component interaction system [19, 20]. Together, these studies establish that degradation prediction benefits from modeling beyond the degrader molecule alone.

A central open question remains: how should unlabeled PROTAC records be leveraged? One natural strategy is to assign pseudo-labels to unlabeled examples and train on the expanded set, as in semi-supervised degradation prediction [18]. While effective in practice, this approach treats unlabeled data as though the primary missing piece were a degradation label. Such a framing is not the only way to exploit these records. The most reliable signal in an unlabeled PROTAC record is not a latent binary activity label but the recorded molecule–target–E3 context itself. Even without a measured degradation value, an entry typically specifies a PROTAC molecule, a target protein, and an E3 ligase. We interpret this tuple as observed design-context evidence: it indicates that a molecule was recorded or studied in relation to a target and an E3 recruiter, but it does not by itself imply active degradation, ternary-complex formation, or biological feasibility.

This observation motivates a different form of self-supervision. Rather than asking a model to impute a degradation label, we ask it to distinguish observed molecule–target–E3 records from counterfactual records formed by replacing the target, the E3 ligase, or both. The model thereby learns a tuple-conditioned pretraining signal: which recorded tuples receive higher scores than sampled alternatives under the chosen corruption process. This idea is connected to the broader principle of learning representations from contrastive structure in unlabeled data [21, 22]. In molecular representation learning, contrastive objectives have been applied to learn robust graph-level or molecular representations [23, 24]. However, PROTAC degradation demands a more specific signal than molecular invariance alone. A given molecule may serve as an effective degrader in one target–E3 context yet fail in another. The self-supervised signal should therefore be defined at the level at which PROTAC records are specified and degradation labels are ultimately assigned: the molecule–target–E3 tuple.

We introduce **DegradeQuery**, a single-model framework built around this principle. DegradeQuery first learns from unlabeled PROTAC records through counterfactual tuple pretraining (CTP). Given an observed tuple (m_i, t_i, e_i), where m_i denotes the PROTAC molecule, t_i the target protein, and e_i the E3 ligase, the model scores this tuple against counterfactual alternatives, including (m_i, t_j, e_i), (m_i, t_i, e_j), and (m_i, t_j, e_k), in which one or both biological components are replaced. This objective trains the representation on molecule–target–E3 records before any degradation label is introduced. The resulting pretraining score is a tuple plausibility score relative to sampled counterfactuals, not a degradation probability. The pretrained model is then fine-tuned on labeled degradation data for binary (active versus inactive) classification. DegradeQuery does not generate pseudo-labels for

unlabeled records and does not rely on teacher models, distillation, or ensembles. Unlabeled data enter the learning process solely through their observed tuple structure.

This design redefines the role of unlabeled PROTAC data. DegradeQuery treats label-missing records as structured tuple observations and uses their recorded molecule–target–E3 context directly during pretraining. The distinction matters because degradation labels are sparse and assay-dependent, whereas the tuple itself is usually available and reflects the molecule, target, and E3 components required by the induced-proximity mechanism [2, 4]. Generic molecular self-supervision can produce robust molecular representations, yet it cannot, on its own, encode which target and E3 contexts accompany a molecule in the observed PROTAC records. Counterfactual tuple pretraining addresses this gap by situating the molecule within the recorded target–E3 context before supervised degradation labels are used.

We evaluate DegradeQuery on the PROTAC-8K benchmark and in additional settings that probe the source and scope of its improvement. On the official split used by DegradeMaster, DegradeQuery achieves the highest reported AUROC and accuracy among the compared methods without relying on pseudo-labels, teacher models, distillation, or ensembles. Matched runs show that the gain is stable on this split, persists when every labeled row is removed from pretraining, and remains present with frozen ESM2-650M protein representations [25]. Component ablations show that tuple pretraining contributes beyond molecule-only self-supervision, while repeated target–E3 group holdouts characterize transfer under changes in biological context. Throughout the paper, tuple plausibility denotes a sampler-relative score learned from recorded tuples and counterfactual replacements; it does not constitute evidence of ternary-complex formation, degradative activity, or biological feasibility.

This work makes three contributions:

(a) We formulate unlabeled PROTAC records as structured molecule–target–E3 observations rather than incomplete examples awaiting pseudo-labels.

(b) We propose DegradeQuery, which employs counterfactual tuple pretraining to learn tuple-conditioned PROTAC representations without assigning degradation pseudo-labels, using teacher models, applying distillation, or ensembling models.

(c) We isolate the contribution of label-missing records and verify the gains from tuple pretraining with frozen ESM2-650M representations. We test scaffold holdout, repeated target–E3 group holdout, and target-wise few-shot adaptation.

2 Related Work

Context-Aware PROTAC Degradation Prediction. Computational approaches to PROTAC degradation prediction have increasingly moved beyond molecule-only representations toward context-aware modeling. DeepPROTACs showed that targeted degradation can be predicted by jointly modeling the PROTAC molecule, the protein of interest, and the E3 ligase [15]. PROTAC-Degradation-Predictor further incorporated E3, target, and assay context into degradation prediction [16]. More recent models make this contextual structure more explicit: PROTAC-STAN uses a structure-informed ternary attention framework to model PROTAC–target–E3 interactions [17], while DegradeMaster introduces a semi-supervised E(3)-equivariant graph neural network with memory-based pseudo-labeling for unlabeled PROTAC records [18]. Related progress in ternary complex prediction and biomolecular interaction modeling similarly suggests that the relevant computational object is often an interaction system rather than an isolated molecule [19, 20]. These studies motivate context-aware degradation prediction; our work differs mainly in how unlabeled records are used, treating them as observed molecule–target–E3 tuples rather than as examples requiring imputed degradation labels.

Learning from Unlabeled Molecular Data. Self-supervised learning has become a standard approach for extracting representations from unlabeled data. Contrastive Predictive Coding and SimCLR learn by distinguishing related views from unrelated ones [21, 22]. In graph and molecular learning, GraphCL uses graph augmentations to learn invariant graph representations [23], and MolCLR applies contrastive learning to molecular graphs for downstream molecular property prediction [24]. These methods show that unlabeled data can provide useful training signals without task-specific labels. For PROTACs, however, molecule-level invariance alone is limited, because the same molecule can be relevant in one target–E3 context and uninformative in another. This motivates defining

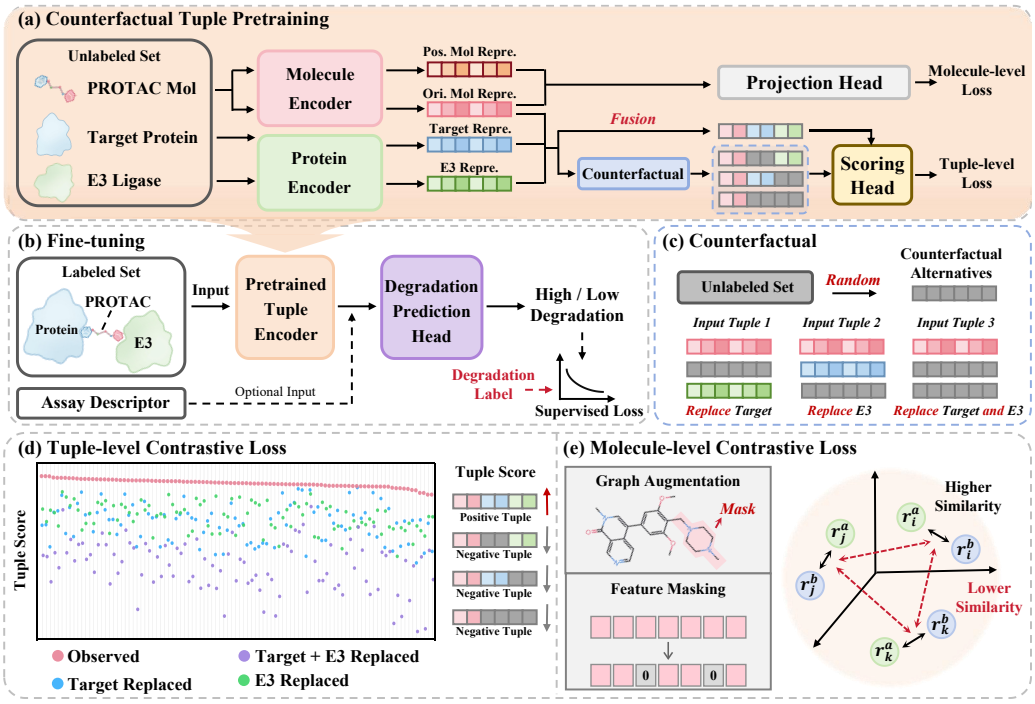


Figure 1: **Overview of DegradeQuery.** DegradeQuery treats the molecule–target–E3 tuple as the unit of unlabeled supervision. In counterfactual tuple pretraining, observed PROTAC tuples from labeled and unlabeled training records are contrasted with alternatives formed by replacing the target, the E3 ligase, or both. This stage learns a tuple-conditioned representation without assigning pseudo-labels to unlabeled records. The pretrained encoder is then fine-tuned on labeled degradation data for binary high/low prediction.

self-supervision at the molecule–target–E3 tuple level rather than only over augmented views of an isolated molecule.

Low-Resource Molecular Prediction. Low-resource learning remains a central challenge in molecular prediction, where many tasks contain only a small number of measured compounds. FS-Mol formalized this setting by aggregating activity measurements across diverse protein targets for few-shot molecular learning [26]. Subsequent work has emphasized the role of task context: Property-Aware Relation Networks learn property-aware molecular embeddings and adaptive relation graphs [27], while context-enriched molecule representations use reference or support molecules to improve few-shot drug discovery [28]. These studies indicate that molecular representations can depend strongly on the task or support context. In this paper, target-wise few-shot adaptation is used as an evaluation protocol for target-conditioned degradation representations, rather than as a claim of a new general-purpose meta-learning algorithm.

3 Method

Overview. Figure 1 summarizes the DegradeQuery workflow. Rather than treating unlabeled PROTAC records as examples with missing degradation labels, DegradeQuery treats them as recorded molecule–target–E3 contexts. It first learns a tuple-conditioned representation by contrasting recorded tuples with sampled target and E3 replacements, and then fine-tunes the resulting encoder on labeled high/low degradation data. The pretraining objective is defined relative to the train-side tuple set and specified counterfactual sampler; it does not imply degradation activity, ternary-complex formation, or biological feasibility.

3.1 Problem Formulation

For a given data split, let $\mathcal{D}_L$ and $\mathcal{D}_U$ denote the train-side labeled and unlabeled subsets, respectively. Each PROTAC record comprises a degrader molecule, a target protein, an E3 ligase, and, optionally, an assay descriptor such as a cell line [9, 10]. Only a subset of records carries a degradation label [18]. We denote the labeled and unlabeled sets as

$$\mathcal{D}_L = \{(m_i, t_i, e_i, c_i, y_i)\}_{i=1}^{n_L}, \qquad \mathcal{D}_U = \{(m_i, t_i, e_i, c_i)\}_{i=1}^{n_U}, \tag{1}$$

where m_i is the PROTAC molecule, t_i is the target protein, e_i is the E3 ligase, c_i is an optional assay descriptor, and $y_i \in \{0, 1\}$ is the binary degradation label.

The goal is to learn a tuple-conditioned degradation predictor

$$p_\theta(y_i = 1 \mid m_i, t_i, e_i, c_i) = \sigma(g_\theta(m_i, t_i, e_i, c_i)), \tag{2}$$

where $\sigma(\cdot)$ denotes the sigmoid function. The unlabeled set is not treated as a collection of molecules with missing labels, but rather as a set of recorded molecule–target–E3 contexts. We use their empirical co-occurrence structure as a self-supervised signal, while recognizing that this structure may reflect literature coverage, database construction, and medicinal-chemistry design bias rather than degradation activity.

For each data split, pretraining constructs the train-side tuple set

$$\mathcal{T} = \{(m_i, t_i, e_i) : (m_i, t_i, e_i, c_i, y_i) \in \mathcal{D}_L\} \cup \{(m_i, t_i, e_i) : (m_i, t_i, e_i, c_i) \in \mathcal{D}_U\}. \tag{3}$$

Held-out test, query, or adaptation-query rows are excluded from $\mathcal{T}$ before pretraining. Degradation labels are not used in this stage. Both active and inactive labeled records enter $\mathcal{T}$ in the same way. In this stage, a positive tuple means that the molecule–target–E3 context was recorded in the dataset; it does not mean that the tuple is degradation-active.

3.2 Tuple-Conditioned Degrader Model

DegradeQuery encodes the molecule, target protein, and E3 ligase into separate representations:

$$h_i^m = f_m(m_i), \qquad h_i^t = f_p(t_i), \qquad h_i^e = f_p(e_i). \tag{4}$$

Here f_m combines a three-layer residual message-passing network over 15-dimensional atom features with a 2,048-bit Morgan fingerprint (radius 2); the two branches are projected to 256 dimensions and summed with layer normalization. The shared protein encoder f_p maps 150 sequence features—amino acid composition, sequence-length and unknown-residue features, and 128 hashed dipeptide and tripeptide counts—to 256 dimensions. The target and E3 representations remain separate after encoding because they serve distinct roles within the tuple.

The fusion module exposes both component identities and pairwise interactions:

$$z_i = \phi_\theta\big(h_i^m \| h_i^t \| h_i^e \| h_i^m \odot h_i^t \| h_i^m \odot h_i^e \| h_i^t \odot h_i^e \| |h_i^t - h_i^e|\big), \tag{5}$$

where $\|$ denotes concatenation and $\odot$ denotes elementwise multiplication. The three products and the target–E3 difference encode pairwise relationships among the molecule, target, and E3 representations. This role-conditioned fusion is shared by the supervised baseline (Sup) and CTP; CTP changes the learned initialization through pretraining while keeping the downstream predictor fixed. When the assay descriptor is available, it is incorporated via $\tilde{z}_i = \psi_\theta(z_i, c_i)$. The self-supervised tuple objective operates on z_i, whereas the downstream degradation classifier operates on $\tilde{z}_i$. We use c_i only in supervised fine-tuning because assay descriptors can be missing or inconsistently annotated for unlabeled records, whereas molecule, target, and E3 identifiers are available for tuple pretraining.

For counterfactual tuple pretraining, DegradeQuery assigns a scalar sampler-relative tuple plausibility score to each molecule–target–E3 tuple:

$$s_\theta(m_i, t_i, e_i) = q_\theta(\phi_\theta(f_m(m_i), f_p(t_i), f_p(e_i))), \tag{6}$$

where q_θ is a scoring head. This score is used exclusively during pretraining. It is meaningful only relative to the sampled counterfactual distribution and is not interpreted as a degradation probability or as an absolute biological compatibility score.

3.3 Counterfactual Tuple Pretraining (CTP)

For each recorded tuple $u_i^+ = (m_i, t_i, e_i)$, we construct counterfactual alternatives by replacing the target, the E3 ligase, or both. For each u_i^+, the sampler draws R valid counterfactuals from the empirical train-side target and E3 pools under the chosen replacement policy. Let $\tilde{t}_{i,r} \neq t_i$ and $\tilde{e}_{i,r} \neq e_i$ denote replacement components. The counterfactual set is

$$\begin{aligned} \mathcal{N}_i &= \{u_{i,r}^-\}_{r=1}^R, \\ u_{i,r}^- &\in \left\{(m_i, \tilde{t}_{i,r}, e_i),\ (m_i, t_i, \tilde{e}_{i,r}),\ (m_i, \tilde{t}_{i,r}, \tilde{e}_{i,r})\right\}. \end{aligned} \tag{7}$$

In split-defined holdout experiments, samples that would create a held-out target–E3 pair are rejected, so held-out pairs are not introduced as corrupted pretraining alternatives.

These counterfactuals are not assumed to be biologically impossible or degradation-inactive. They are sampler-defined alternatives that may include untested but viable molecule–target–E3 combinations. Thus, the objective is noise-contrastive rather than supervised with true negative biological examples: the recorded tuple is encouraged to score above sampled alternatives under the chosen replacement distribution.

The tuple scoring function is optimized with a contrastive ranking loss:

$$\mathcal{L}_{\text{tuple}} = -\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \log \frac{\exp\big(s_\theta(u_i^+)/\tau_{\text{tuple}}\big)}{\exp\big(s_\theta(u_i^+)/\tau_{\text{tuple}}\big) + \sum_{u^- \in \mathcal{N}_i} \exp\big(s_\theta(u^-)/\tau_{\text{tuple}}\big)}, \tag{8}$$

where $\mathcal{B}$ is a minibatch and τ_{tuple} is the tuple temperature. This loss relies solely on train-side recorded tuples in $\mathcal{T}$ and does not require, infer, or assign degradation labels.

In addition, following graph and molecular contrastive learning, we employ a lightweight molecular consistency objective to encourage augmentation-invariant molecular representations [23, 24]. We generate two stochastic molecular graph views $a(m_i)$ and $b(m_i)$ using atom-feature masking and edge dropping. Let f_m^{graph} denote the graph branch of the molecule encoder, and define

$$r_i^a = \rho_\theta(f_m^{\text{graph}}(a(m_i))), \qquad r_i^b = \rho_\theta(f_m^{\text{graph}}(b(m_i))), \tag{9}$$

where ρ_θ is a projection head. The directional normalized temperature-scaled contrastive loss is [22]:

$$\mathcal{L}_{\text{mol}}^{a \to b} = -\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \log \frac{\exp\big(\text{sim}(r_i^a, r_i^b)/\tau_{\text{mol}}\big)}{\sum_{j \in \mathcal{B}} \exp\big(\text{sim}(r_i^a, r_j^b)/\tau_{\text{mol}}\big)}, \tag{10}$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity and τ_{mol} is the molecule temperature. Training uses the symmetric objective $\mathcal{L}_{\text{mol}} = (\mathcal{L}_{\text{mol}}^{a \to b} + \mathcal{L}_{\text{mol}}^{b \to a})/2$.

The full pretraining objective combines both terms:

$$\mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{tuple}} + \lambda_{\text{mol}} \mathcal{L}_{\text{mol}}. \tag{11}$$

The tuple loss is the primary training signal, while the molecule-level term serves as an auxiliary regularizer.

3.4 Fine-Tuning and Prediction

After pretraining, the tuple encoder is fine-tuned on the labeled degradation records. The model predicts

$$\hat{p}_i = p_\theta(y_i = 1 \mid m_i, t_i, e_i, c_i) = \sigma\big(w^\top \tilde{z}_i + b\big), \tag{12}$$

where $\tilde{z}_i$ is the descriptor-aware representation.

We optimize a binary classification loss over labeled examples only. Let p_i^* denote the predicted probability assigned to the true class:

$$p_i^* = y_i \hat{p}_i + (1 - y_i)(1 - \hat{p}_i). \tag{13}$$

The supervised objective is a class-weighted focal loss [29]:

$$\mathcal{L}_{\text{sup}} = -\frac{1}{|\mathcal{D}_L|} \sum_{i \in \mathcal{D}_L} \alpha_{y_i} (1 - p_i^*)^\gamma \log p_i^*. \tag{14}$$

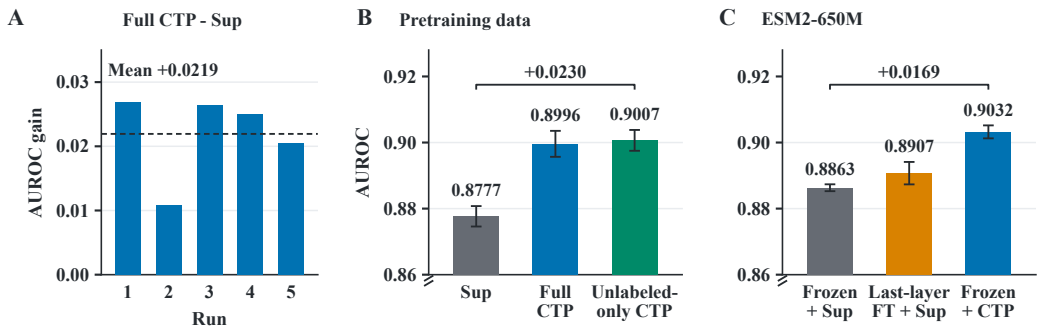


Figure 2: CTP gains are reproducible and recoverable from label-missing records. (A) Full CTP improves AUROC in all five paired official-split runs; the dashed line marks the mean gain of 0.0219. (B) Unlabeled-only CTP retains the gain using 7,134 label-missing records and zero labeled records during pretraining. (C) Frozen ESM2-650M representations also benefit from CTP inside the same tuple predictor. Brackets mark the unlabeled-only CTP − Sup difference in (B) and the frozen CTP − Sup difference in (C). Bars and error bars in (B,C) show means and standard deviations over five and three runs, respectively. Their expanded AUROC axes mark the omitted lower range.

Setting $\gamma = 0$ recovers class-weighted binary cross-entropy.

The two training stages can be summarized as $\theta_{\text{pre}} = \arg\min_\theta \mathcal{L}_{\text{pre}}(\mathcal{T})$, followed by

$$\theta^\star = \arg\min_\theta \mathcal{L}_{\text{sup}}(\mathcal{D}_L), \qquad \theta \text{ initialized from } \theta_{\text{pre}}. \tag{15}$$

Unlike pseudo-label-based semi-supervised PROTAC degradation prediction [18], unlabeled records contribute exclusively through the tuple pretraining objective: they are used as recorded contexts, never as degradation-labeled examples. At inference time, DegradeQuery applies the fine-tuned model directly: $\hat{y}_i = \mathbb{I}[\hat{p}_i \geq \delta]$, where $\mathbb{I}[\cdot]$ is the indicator function and δ is a fixed decision threshold specified by the evaluation protocol. No teacher model, pseudo-labeling step, distillation procedure, or model ensemble is used.

4 Experiments

We evaluate whether CTP improves performance on PROTAC-8K without pseudo-labeling, whether label-missing records alone account for the gain, whether CTP remains useful with a strong protein representation, and how performance changes under target–E3 group holdout, scaffold holdout, and target-wise few-shot adaptation.

4.1 Experimental Setup

We evaluate binary high/low PROTAC degradation prediction on the PROTAC-8K benchmark used in prior work [18]. The normalized dataset contains 8,636 molecule–target–E3 records, of which 1,502 are labeled and 7,134 are unlabeled. Molecule, target, and E3 identifiers are available for all records, so PROTAC-8K is label-sparse but tuple-rich enough for self-supervised tuple pretraining. All experiments are split-specific. Test or query rows are never used for tuple pretraining, supervised fine-tuning, model selection, or threshold tuning. In target–E3 group and scaffold holdouts, held-out records are excluded from both tuple pretraining and supervised training. For target–E3 group holdout, the sampler also rejects corrupted tuples whose target–E3 pair is held out, preventing held-out biological pairs from entering pretraining as negatives. In target-wise few-shot adaptation, support rows are used only for adaptation and query rows only for evaluation; neither is used during base pretraining or base supervised training. We compare DegradeQuery-Sup, trained only on labeled training records, with DegradeQuery-CTP, which first pretrains on train-side molecule–target–E3 tuples, including unlabeled records, and then fine-tunes on labeled records. CTP uses no pseudo-labels, teacher models, distillation, or ensembles. AUROC is the primary metric, accuracy is reported for benchmark compatibility, and threshold-dependent metrics use a fixed threshold of 0.5. Repeated-run statistics and group-holdout variability are detailed in Appendix A.5. Metrics are reported on the 0–1 scale.

4.2 PROTAC-8K Benchmark

Table 1 compares DegradeQuery with prior baselines under the official PROTAC-8K protocol used by DegradeMaster [18]. Unlike DegradeMaster-Semi, which uses pseudo-labeling, DegradeQuery-CTP uses unlabeled records only as molecule–target–E3 tuple observations. CTP achieves the best accuracy and AUROC, improving over DegradeMaster-Semi by 0.0134 accuracy and 0.0240 AUROC, and over DegradeQuery-Sup by 0.0367 accuracy and 0.0238 AUROC. This supports the use of unlabeled PROTAC records as tuple-level supervision rather than merely as pseudo-label candidates. Figure 2 and Table 2 provide matched reliability and attribution analyses. Every paired official-split run shows an AUROC improvement for CTP over Sup; the mean paired gain is 0.0219. The same runs also improve area under the precision–recall curve (AUPRC), accuracy, and Matthews correlation coefficient (MCC). Full metrics and paired uncertainty estimates are reported in Appendix A.5.

4.3 Isolating the Label-Missing Signal

We remove every labeled row from the pretraining set while keeping the supervised fine-tuning rows and official test split fixed. This unlabeled-only variant pretrains on 7,134 label-missing records and zero labeled records. It reaches 0.9007 ± 0.0032 AUROC, improving over Sup by 0.0230. Its mean AUROC exceeds Full CTP by 0.0011. The paired comparison in Appendix A.5 shows that this difference is within the uncertainty of the repeated runs. Label-missing records therefore retain the benefit of tuple pretraining without using labeled rows during that stage.

We also replace the original protein features with ESM2-650M target and E3 representations inside the same DegradeQuery tuple predictor. With frozen ESM2 representations, AUROC increases from 0.8863 for Sup to 0.9032 for CTP. CTP also exceeds the supervised reference that fine-tunes the final ESM2 layer together with the projection and prediction layers (0.8907). ESM2 supplies the component representations in these experiments, while the complete molecule–target–E3 predictor and role-conditioned fusion remain unchanged. The matched architecture and split isolate the source of the improvement: unlabeled-only CTP removes labeled-row reuse, while the ESM2 comparison shows that tuple pretraining supplies information beyond a strong protein representation.

Table 1: **Official PROTAC-8K benchmark comparison.** Prior baseline values use the official protocol reported by DegradeMaster; accuracy values originally reported as percentages are converted to the 0–1 scale. DegradeQuery-CTP uses unlabeled records through tuple pretraining without activity pseudo-labels, teacher models, distillation, or ensembles.

Method	Unlabeled signal	Accuracy	AUROC
DeepPROTACs	None	0.5821	0.7107
DeepPROTACs	Semi-supervised	0.5902	0.7171
PROTAC-STAN	None	0.7934	0.7986
PROTAC-STAN	Semi-supervised	0.7825	0.7896
DegradeMaster	None	0.8141	0.8541
DegradeMaster	Pseudo-labeling	0.8366	0.8825
DegradeQuery-Sup	None	0.8133	0.8827
DegradeQuery-CTP	Tuple pretraining	**0.8500**	**0.9065**

Table 2: Controlled attribution on the official PROTAC-8K split. Values are mean $\pm$ standard deviation. All variants remain full PROTAC tuple predictors; in the ESM2 block, ESM2 supplies the target and E3 representations. FT denotes fine-tuning.

Protein representation	Training	AUROC	Accuracy
Original	Sup ($n = 5$)	0.8777±0.0031	0.7840±0.0072
Original	Full CTP ($n = 5$)	0.8996±0.0039	0.8233±0.0058
Original	Unlabeled-only CTP ($n = 5$)	**0.9007±0.0032**	**0.8267±0.0097**
Frozen ESM2-650M	Sup ($n = 3$)	0.8863±0.0010	0.7933±0.0067
ESM2-650M last-layer FT	Sup ($n = 3$)	0.8907±0.0034	0.8144±0.0150
Frozen ESM2-650M	CTP ($n = 3$)	**0.9032±0.0020**	**0.8322±0.0107**

4.4 Pretraining Ablation

We next compare the tuple objective with generic molecule-level self-supervision. Molecule-only self-supervised learning (SSL) is a meaningful baseline because graph and molecular contrastive learning have been effective for molecular representation learning [23, 24]. Table 3 compares the pretraining components and includes the supervised benchmark reference. Molecule-only SSL improves over labeled-only training, while tuple-only pretraining gives the higher AUROC and AUPRC of the two individual components. Combining the objectives gives the highest AUROC, showing that the tuple objective contributes beyond generic molecular consistency.

Table 3: Component analysis on the PROTAC-8K split. Pretraining variants are averaged over repeated runs; Sup is the supervised benchmark reference. Replication details are given in Appendix A.3.

Pretraining	AUROC	AUPRC	Accuracy
Sup (benchmark)	0.8827	0.8200	0.8133
Molecule-only SSL	0.8910	0.7947	**0.8289**
Tuple-only pretraining	0.8943	**0.8432**	0.8056
Full CTP	**0.8991**	0.8386	0.8256

As a shortcut control, a molecule-ablated model using only target, E3, and assay-side features obtains AUROC near 0.55 and MCC near zero across repeated runs. Across six E3-only, target-only, joint target–E3, frequency-matched, mixed, and hard-mixed policies, AUROC ranges from 0.8932 to 0.9088; every policy exceeds the supervised benchmark reference of 0.8827 (Appendix A.5). The gain persists with frequency-matched candidates and with individual or joint component replacement. Across the tested tuple temperature, molecule temperature, and molecule-loss weight settings, AUROC remains between 0.8991 and 0.9041; the exact sensitivity values appear in Appendix A.5. Together, these controls establish that the tuple contribution persists across the tested sampling and optimization choices.

4.5 Conditional Generalization and Few-Shot Adaptation

We finally test whether tuple pretraining remains useful when the degradation context changes. We evaluate repeated target–E3 group holdout, scaffold holdout, and target-wise few-shot adaptation, following the broader motivation of context-aware PROTAC and ternary-complex modeling [17–19]. Table 4 reports the results.

Table 4: Selected generalization and target-wise few-shot adaptation results. Target–E3 rows average independently constructed group holdouts; other rows average repeated runs. Replication details are given in Appendix A.8. Few-shot rows use K positive and K negative support examples per held-out target and evaluate the remaining labeled rows. Bold indicates the better method within each setting and metric.

Setting	Method	AUROC	AUPRC	MCC	BalAcc
Target–E3 group holdout	Sup	0.6769	0.5964	0.2810	0.6235
Target–E3 group holdout	CTP	**0.7064**	**0.6110**	**0.2947**	**0.6362**
Scaffold holdout	Sup	0.8681	0.8236	0.5919	0.7884
Scaffold holdout	CTP	**0.8858**	**0.8481**	**0.6358**	**0.8112**
Target-wise adaptation, $K = 2$	Sup	0.6686	0.6731	0.1263	0.5765
Target-wise adaptation, $K = 2$	CTP	**0.6812**	**0.6847**	**0.3521**	**0.6867**
Target-wise adaptation, $K = 4$	Sup	0.6673	0.6248	0.2681	0.6381
Target-wise adaptation, $K = 4$	CTP	**0.7173**	**0.6692**	**0.3146**	**0.6700**
Target-wise adaptation, $K = 8$	Sup	0.7790	0.7727	0.4196	0.7005
Target-wise adaptation, $K = 8$	CTP	**0.7811**	**0.7976**	**0.4223**	**0.7200**

The official test contains 283 labeled-seen target–E3 examples, 16 seen only during train-side pretraining, and one fully unseen pair; on the 283 labeled-seen examples, AUROC increases from 0.8923 for Sup to 0.9130 for CTP. As the stronger context-shift test, repeated target–E3 group holdout excludes complete target–recruiter groups from both training stages and rejects them during

counterfactual sampling. Across independently constructed 20% group holdouts, mean AUROC increases from 0.6769 to 0.7064. The gain varies with the held-out target–E3 groups (Appendix A.5). On the original grouped split, AUROC increases from 0.6501 to 0.7118 and accuracy from 0.6589 to 0.6744. Its test set is concentrated in target P10275 (180/300 examples) and E3 ligase Q96SW2 (172/300). Targets O43924 and P00533 have ten examples each and positive rates of 0.10. Repeated class-balanced holdouts reduce dependence on this composition while preserving the positive mean CTP advantage. Scaffold holdout tests chemical generalization by excluding held-out Bemis–Murcko scaffold groups from base pretraining and supervised training. CTP improves AUROC from 0.8681 to 0.8858 and MCC from 0.5919 to 0.6358, with positive gains across all reported metrics. These results demonstrate transfer to held-out chemical scaffolds. Target-wise few-shot adaptation evaluates initialization quality rather than proposing a general meta-learning algorithm. For each held-out target, we adapt on a balanced support set with K positive and K negative examples and evaluate on the remaining labeled query rows. Across $K = 2$, $K = 4$, and $K = 8$, CTP improves every reported metric; the largest fixed-threshold gain appears at $K = 2$ in MCC, increasing from 0.1263 to 0.3521. Because changing K changes the remaining query set, the three settings should be read as separate adaptation protocols rather than a monotonic learning curve.

Overall, unlabeled molecule–target–E3 tuples provide useful supervision beyond the main benchmark. The benefit is stable on the official split, remains positive on average under target–E3 group holdout, and improves the reported scaffold and target-wise few-shot settings.

4.6 Diagnostic Analyses and Evidence Scope

We shuffle target–E3 assignments during pretraining while holding molecule records, component frequencies, training size, downstream split, and optimization fixed. Observed-tuple and shuffled-pair CTP obtain AUROCs of 0.8991 and 0.9021, with a mean paired difference of 0.0029. The paired analysis in Appendix A.5 does not establish a distinct benefit from preserving exact observed target–E3 assignments.

Train-side temperature scaling reduces expected calibration error (ECE) from 0.0703 to 0.0626 for Sup and from 0.0994 to 0.0772 for CTP; negative log-likelihood (NLL) changes from 0.4300 to 0.4237 and from 0.4671 to 0.4357, respectively. The scalar temperature is fit only on train-side calibration data and fixed for test evaluation; scaling improves both models without establishing an intrinsic calibration advantage for CTP and leaves AUROC unchanged. An external-source audit further finds that the available PROTAC-Bench release shares 2,461 canonical SMILES and 156 targets with PROTAC-8K and lacks sample-level DC_{50} and $D_{\max}$ values required for matched relabeling. These properties confound a direct external comparison through source overlap, endpoint conversion, and split construction.

5 Discussion and Conclusion

DegradeQuery derives supervision from relational units in label-missing PROTAC records. Counterfactual replacement organizes molecule–target–E3 contexts before degradation labels are introduced, and the signal remains useful with ESM2 representations. Role-conditioned fusion retains distinct molecule–target, molecule–E3, and target–E3 associations.

The learned tuple score is relative to the recorded database and replacement distribution and does not recover ternary-complex geometry, cooperativity, binding kinetics, ubiquitination efficiency, or cellular exposure. Target–E3 transfer depends on biological coverage, while database curation and publication frequency shape the available tuples. Prospective evaluation therefore requires deduplicated compounds, harmonized endpoints, predeclared splits, and broader target–E3 coverage. We provide code, experiment configurations, fixed splits, and reproduction instructions.

Controlled ablations locate the benefit: tuple-only pretraining exceeds molecule-only SSL, the combined objective yields the highest AUROC, and unlabeled-only CTP preserves the improvement over Sup. CTP also improves over Sup with frozen ESM2-650M representations inside the same predictor. Recorded tuple context therefore complements molecular and protein representations and remains useful under scaffold shift and target-wise few-shot adaptation.

Acknowledgements and Disclosures

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62476177 and 62471310), the Guangdong Natural Science Foundation Project (Grant No. 2025A1515011567), and the Shenzhen Medical Research Fund (Grant No. D2601008). This work also received support from the National Natural Science Foundation Major Scientific Research Instrument Development Project (Grant No. 62527809) and the Shenzhen Key Industry R&D Program Project (Grant No. ZDCY20250901102300001). We thank the Intelligent Computing Center of Shenzhen University for computational support.

The authors declare no competing interests.

References

[1] Kathleen M Sakamoto, Kyung B Kim, Akiko Kumagai, Frank Mercurio, Craig M Crews, and Raymond J Deshaies. PROTACs: Chimeric molecules that target proteins to the Skp1–Cullin–F box complex for ubiquitination and degradation. *Proceedings of the National Academy of Sciences*, 98(15):8554–8559, 2001.

[2] Ashton C Lai and Craig M Crews. Induced protein degradation: an emerging drug discovery paradigm. *Nature Reviews Drug Discovery*, 16(2):101–114, 2017.

[3] George M Burslem and Craig M Crews. Proteolysis-targeting chimeras as therapeutics and tools for biological discovery. *Cell*, 181(1):102–114, 2020.

[4] Miklós Békés, David R Langley, and Craig M Crews. PROTAC targeted protein degraders: the past is prologue. *Nature Reviews Drug Discovery*, 21(3):181–200, 2022.

[5] Daniel P Bondeson, Alina Mares, Ian ED Smith, Eunhwa Ko, Sebastien Campos, Afjal H Miah, Katie E Mulholland, Natasha Routly, Dennis L Buckley, Jeffrey L Gustafson, et al. Catalytic in vivo protein knockdown by small-molecule PROTACs. *Nature Chemical Biology*, 11(8): 611–617, 2015.

[6] Georg E Winter, Dennis L Buckley, Joshiawa Paulk, Justin M Roberts, Amanda Souza, Sirano Dhe-Paganon, and James E Bradner. Phthalimide conjugation as a strategy for in vivo target protein degradation. *Science*, 348(6241):1376–1381, 2015.

[7] Jing Lu, Yimin Qian, Martha Altieri, Hanqing Dong, Jing Wang, Kanak Raina, John Hines, James D Winkler, Andrew P Crew, Kevin Coleman, et al. Hijacking the E3 ubiquitin ligase cereblon to efficiently target BRD4. *Chemistry & Biology*, 22(6):755–763, 2015.

[8] Matthieu Schapira, Matthew F Calabrese, Alex N Bullock, and Craig M Crews. Targeted protein degradation: expanding the toolbox. *Nature Reviews Drug Discovery*, 18(12):949–963, 2019.

[9] Gaoqi Weng, Chao Shen, Dongsheng Cao, Junbo Gao, Xiaowu Dong, Qiaojun He, Bo Yang, Dan Li, Jian Wu, and Tingjun Hou. PROTAC-DB: an online database of PROTACs. *Nucleic Acids Research*, 49(D1):D1381–D1387, 2021.

[10] Gaoqi Weng, Xuanyan Cai, Dongsheng Cao, Hongyan Du, Chao Shen, Yafeng Deng, Qiaojun He, Bo Yang, Dan Li, and Tingjun Hou. PROTAC-DB 2.0: an updated database of PROTACs. *Nucleic Acids Research*, 51(D1):D1367–D1372, 2023.

[11] Hong Cai, Gengyuan Yao, Yulong Shi, Tianyi Zhang, and Yuanjia Hu. PROTAC-PatentDB: A PROTAC patent compound dataset. *Scientific Data*, 12(1):1840, 2025.

[12] Jingxuan Ge, Shimeng Li, Gaoqi Weng, Huating Wang, Meijing Fang, Huiyong Sun, Yafeng Deng, Chang-Yu Hsieh, Dan Li, and Tingjun Hou. PROTAC-DB 3.0: an updated database of PROTACs with extended pharmacokinetic parameters. *Nucleic Acids Research*, 53(D1): D1510–D1515, 2025.

[13] Xiao Wang, Zhiyao Zhuang, Chengwei Zhang, Bowen Zhang, Wei Zhan, Yifan Wang, Zhaojuan Liu, Shanwen Yuan, Wenjia Niu, Qi He, et al. MolGlueDB: an online database of molecular glues. *Nucleic Acids Research*, 54(D1):D1510–D1518, 2026.

[14] Jintao Zhu, Yiyan Liao, Haoyu Lin, Juan Xie, Zhichao Deng, Jinyu Han, Zhen Zhang, Jinchuan Xiao, Zhiyao Wang, Shuaipeng Zhang, et al. MGTbind: a comprehensive database of molecular glue ternary interactome. *Nucleic Acids Research*, 54(D1):D1500–D1509, 2026.

[15] Fenglei Li, Qiaoyu Hu, Xianglei Zhang, Renhong Sun, Zhuanghua Liu, Sanan Wu, Siyuan Tian, Xinyue Ma, Zhizhuo Dai, Xiaobao Yang, et al. DeepPROTACs is a deep learning-based targeted degradation predictor for PROTACs. *Nature Communications*, 13(1):7133, 2022.

[16] Stefano Ribes, Eva Nittinger, Christian Tyrchan, and Rocío Mercado. Modeling PROTAC degradation activity with machine learning. *Artificial Intelligence in the Life Sciences*, 6:100104, 2024.

[17] Zhenglu Chen, Chunbin Gu, Shuoyan Tan, Xiaorui Wang, Yuquan Li, Mutian He, Ruiqiang Lu, Shijia Sun, Chang-Yu Hsieh, Xiaojun Yao, et al. Interpretable PROTAC degradation prediction with structure-informed deep ternary attention framework. *Advanced Science*, 12(47):e08138, 2025.

[18] Jie Liu, Michael J Roy, Luke Isbel, and Fuyi Li. Accurate PROTAC-targeted degradation prediction with DegradeMaster. *Bioinformatics*, 41(Supplement_1):i342–i351, 2025.

[19] Fanglei Xue, Meihan Zhang, Shuqi Li, Xinyu Gao, James A Wohlschlegel, Wenbing Huang, Yi Yang, and Weixian Deng. SE(3)-equivariant ternary complex prediction towards target protein degradation. *Nature Communications*, 16(1):5514, 2025.

[20] 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.

[21] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[22] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020.

[23] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33:5812–5823, 2020.

[24] Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3): 279–287, 2022.

[25] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

[26] Megan Stanley, John Bronskill, Krzysztof Maziarz, Hubert Misztela, Jessica Lanini, Marwin Segler, Nadine Schneider, and Marc Brockschmidt. FS-Mol: A few-shot learning dataset of molecules. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[27] Yaqing Wang, Abulikemu Abuduweili, Quanming Yao, and Dejing Dou. Property-aware relation networks for few-shot molecular property prediction. *Advances in Neural Information Processing Systems*, 34:17441–17454, 2021.

[28] Johannes Schimunek, Philipp Seidl, Lukas Friedrich, Daniel Kuhn, Friedrich Rippmann, Sepp Hochreiter, and Günter Klambauer. Context-enriched molecule representations improve few-shot drug discovery. In *The Eleventh International Conference on Learning Representations*, 2023.

[29] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

A Technical Appendices

A.1 Dataset, Labels, and Split Protocol

Dataset summary. Table 5 summarizes the normalized PROTAC-8K data used in all experiments. The dataset is label-sparse but tuple-rich: only 1,502 of 8,636 records have binary degradation labels, whereas molecule, target, and E3 identifiers are available for every record. This property motivates the tuple-level pretraining objective used by DegradeQuery-CTP.

Table 5: Summary of the normalized PROTAC-8K dataset.

Item	Count
Total molecule–target–E3 records	8,636
Labeled records	1,502
Unlabeled records	7,134
Unique canonical SMILES	5,656
Unique targets	332
Unique E3 ligases	16
Unique target–E3 pairs	563
Rows with molecule, target, and E3 identifiers	8,636
Molecule–target–E3 identifier coverage	100%

Binary label definition. We use the PROTAC-8K benchmark binary label without introducing any additional relabeling threshold. The released benchmark preprocessing defines low activity using $DC_{50} \geq 100\,\mathrm{nM}$ and $D_{\max} < 80\%$, with the remaining records assigned to the high-activity class. The public benchmark table does not contain sample-level DC_{50} and $D_{\max}$ values, so matched re-binarization under alternative activity thresholds is unavailable from the released data. The label $y = 1$ denotes active/high degradation and $y = 0$ denotes inactive/low degradation under this benchmark preprocessing. The decision threshold $\delta = 0.5$ is used only to convert model probabilities into binary predictions for threshold-dependent metrics such as accuracy, F1, MCC, specificity, and balanced accuracy.

Table 6: Class balance of the labeled official PROTAC-8K split used for supervised fine-tuning and benchmark evaluation.

Split	Active/high ($y = 1$)	Inactive/low ($y = 0$)	Total
Training labeled split	469	733	1,202
Test labeled split	108	192	300
Total labeled records	577	925	1,502

Official-split target–E3 exposure. Among the 300 official-test examples, 283 involve target–E3 pairs seen in labeled training, 16 involve pairs seen only in train-side pretraining records, and one involves a pair absent from both training sources. On the 283 labeled-seen examples, Sup and CTP obtain AUROCs of 0.8923 and 0.9130, respectively. The remaining 17 examples combine pretraining-seen/labeled-unseen pairs with a single fully unseen pair, so pair-disjoint group holdouts provide the statistically meaningful cold-context evaluation reported in Section 4.5.

Split-specific exclusion policy. All experiments are split-specific. Test or query rows are never used for tuple pretraining, supervised fine-tuning, model selection, or threshold tuning. For DegradeQuery-CTP, tuple pretraining is restricted to train-side molecule–target–E3 tuples only. For holdout experiments, held-out records are removed from both tuple pretraining and supervised training. For few-shot adaptation, support rows are used only for adaptation and query rows are used only for evaluation.

Leakage audit. Table 8 reports the leakage audit used for all split-specific experiments. There is no row-level, tuple-level, negative-sampling, query-set, model-selection, threshold-tuning, or pseudo-label leakage. In particular, exact test/query molecule–target–E3 tuples do not appear in the train-side pretraining tuple set, and the pretraining stage never sees exact test/query tuple identities.

Table 7: Split-specific training and evaluation policy. "Train-side tuples" include labeled and unlabeled rows available within the corresponding training split.

Setting	Tuple pretraining	Supervised fine-tuning / adaptation	Evaluation-only rows
PROTAC-8K benchmark	Train-side molecule–target–E3 tuples only	Labeled training rows only	Official test rows
Target–E3 group holdout	Train-side tuples excluding held-out target–E3 pairs	Labeled training rows excluding held-out target–E3 pairs	Held-out target–E3 group rows
Scaffold holdout	Train-side tuples excluding held-out scaffold groups	Labeled training rows excluding held-out scaffold groups	Held-out scaffold rows
Target-wise few-shot	Base train-side tuples only; query rows excluded	Balanced support rows with K positives and K negatives per held-out target	Remaining labeled query rows for each held-out target

Table 8: Leakage audit for tuple-level pretraining and downstream evaluation.

Audit item	Count
Test/query rows used in tuple pretraining	0
Test/query rows used in supervised fine-tuning	0
Test/query rows used for model selection	0
Test/query rows used for threshold tuning	0
Exact test/query tuples present in train-side pretraining tuples	0
Exact test/query tuple identities seen during pretraining	0
Held-out target–E3 pairs sampled as negatives	0
Observed train-side tuples sampled as counterfactual negatives	0
Few-shot query rows used during adaptation	0
Pseudo-labels assigned to unlabeled records	0
Pseudo-labels assigned to test/query records	0

A.2 Model and Pretraining Details

Shared architecture across Sup and CTP. DegradeQuery-Sup and DegradeQuery-CTP use the same tuple-conditioned architecture, the same protein feature pipeline, the same molecular feature pipeline, the same supervised loss, the same labeled training split, the same model-selection rule, and the same fixed decision threshold. The only difference is that DegradeQuery-CTP initializes the encoder from counterfactual tuple pretraining before supervised fine-tuning, whereas DegradeQuery-Sup is trained only with labeled degradation data.

Counterfactual tuple construction. For each observed train-side tuple $u_i^+ = (m_i, t_i, e_i)$, the pretraining sampler constructs counterfactual tuples by replacing the target, the E3 ligase, or both. The mixed replacement setting uses all three mismatch types and corresponds to the full counterfactual tuple pretraining objective used by DegradeQuery-CTP. The default sampler attempts one accepted counterfactual of each type, giving up to three counterfactuals per observed tuple. For target-only replacement it samples from records sharing the original E3; for E3-only replacement it samples from records sharing the original target. Joint replacement draws both components from the train-side pools. Candidates that violate the rejection rules are discarded, and unavailable replacement types are masked in the loss.

No pseudo-labeling. Unlabeled records are used only through their observed molecule–target–E3 tuple structure. No degradation pseudo-label is assigned to any unlabeled record. No teacher model, distillation procedure, or ensemble is used in either pretraining or fine-tuning.

Table 9: Model components used by DegradeQuery.

Component	Description
Molecule encoder f_m	Three residual message-passing layers over 15-dimensional atom features plus a 2,048-bit Morgan fingerprint (radius 2). Both branches are projected to 256 dimensions, summed, and layer-normalized.
Protein encoder f_p	Maps 150 sequence features to 256 dimensions: amino acid composition, sequence-length and unknown-residue features, and 128 hashed dipeptide and tripeptide counts. The encoder is shared, while target and E3 embeddings remain separate.
Tuple fusion module ϕ_θ	Concatenates molecule, target, and E3 embeddings; their three pairwise products; and the absolute target–E3 difference before projection.
Assay-aware module ψ_θ	Adds the available cell-line embedding for downstream prediction; assay context is not used by the tuple pretraining objective.
Compatibility scoring head q_θ	Produces the tuple plausibility score $s_\theta(m, t, e)$ during counterfactual tuple pretraining.
Supervised classifier	Predicts the benchmark binary active/high versus inactive/low degradation label after fine-tuning.

Table 10: Counterfactual tuple construction used in tuple pretraining.

Tuple type	Construction
Observed tuple	(m_i, t_i, e_i) from the train-side tuple set.
Target replacement	$(m_i, \tilde{t}, e_i)$, where $\tilde{t} \neq t_i$.
E3 replacement	$(m_i, t_i, \tilde{e})$, where $\tilde{e} \neq e_i$.
Joint target–E3 replacement	$(m_i, \tilde{t}, \tilde{e})$, where $\tilde{t} \neq t_i$ and $\tilde{e} \neq e_i$.
Mixed replacement	Union of target-only, E3-only, and joint target–E3 replacements.
Sampler rejection rules	Reject exact observed train-side tuples, exact test/query tuples, and held-out target–E3 pairs in the corresponding holdout setting.

Table 11: Counterfactual tuple pretraining configuration.

Item	Value
Optimizer	AdamW
Learning rate	3×10^{-4}
Weight decay	1×10^{-4}
Batch size	64
Pretraining epochs	60
Counterfactual samples per observed tuple	Up to 3
Tuple temperature τ_{tuple}	0.1
Molecule temperature τ_{mol}	0.1
Tuple-loss coefficient λ_{tuple}	1.0
Molecule-loss coefficient λ_{mol}	0.1
Pretraining rows, full CTP	Train-side labeled and label-missing records
Pretraining rows, unlabeled-only CTP	7,134 label-missing records; zero labeled records

A.3 Fine-Tuning Configuration

Table 12 gives the supervised fine-tuning configuration used for both DegradeQuery-Sup and DegradeQuery-CTP. The class weights are computed automatically from the labeled training split class frequency.

Table 12: Supervised fine-tuning configuration.

Item	Value
Optimizer	AdamW
Learning rate	1×10^{-4}
Weight decay	1×10^{-4}
Batch size	64
Fine-tuning epochs	Up to 120
Focal loss γ	1.0
Class weights α_y	Auto; computed from the labeled training split class frequency. Official split: $\alpha_1 = 0.6098$, $\alpha_0 = 0.3902$.
Model selection	Train-side validation only; official test labels are excluded
Decision threshold	0.5
Seeds	Five for official reliability; three for component and ESM2 analyses

Table 13: Reproducibility settings shared by the main experiments.

Item	Setting
Random seeds	1–5 for official reliability; 1–3 for component and ESM2 analyses
Reported values	Mean $\pm$ standard deviation
Threshold-dependent metrics	Fixed threshold 0.5
Calibration	Raw probabilities; no post-hoc calibration
Unlabeled usage in CTP	Tuple pretraining only
Unlabeled usage in Sup	None
Teacher models	None
Distillation	None
Model ensembles	None
Pseudo-labeling	None
Test/query participation in training	None

A.4 Data and Code Availability

The PROTAC-8K benchmark is available from its cited source. The public DegradeQuery repository provides training and evaluation code, preprocessing scripts, experiment configurations, fixed split files, and reproduction instructions.

A.5 Supplementary Results

Five-seed reliability and unlabeled-only control. Table 14 reports the complete official-split comparison used in Section 4.3. All three methods use the same labeled fine-tuning rows and test set. Full CTP uses all available train-side tuples during pretraining, while unlabeled-only CTP removes every labeled row from pretraining.

For Full CTP minus Sup, the paired 95% confidence interval for the AUROC difference is $[0.0136, 0.0303]$. For unlabeled-only CTP minus Sup, the intervals for AUROC, accuracy, AUPRC, and MCC are $[0.0173, 0.0287]$, $[0.0301, 0.0552]$, $[0.0320, 0.0640]$, and $[0.0492, 0.1057]$, respectively. The intervals for balanced accuracy, Brier score, and ECE are $[0.0208, 0.0507]$, $[-0.0261, -0.0169]$, and $[-0.0394, -0.0065]$, respectively.

For unlabeled-only CTP minus Full CTP, the paired 95% confidence intervals for AUROC, accuracy, AUPRC, and MCC are $[-0.0064, 0.0085]$, $[-0.0122, 0.0188]$, $[-0.0097, 0.0101]$, and $[-0.0227, 0.0468]$, respectively. The intervals for balanced accuracy, Brier score, and ECE are $[-0.0100, 0.0273]$, $[-0.0104, 0.0065]$, and $[-0.0125, 0.0114]$, respectively.

Counterfactual construction. Table 15 compares six sampling policies under the same three-seed analysis. Every tested policy yields an AUROC between 0.8932 and 0.9088, so the improvement does not depend on one replacement mixture.

Hyperparameter sensitivity. The principal result is stable over the tested tuple temperature, molecule temperature, and molecule-loss weight ranges (Table 16).

Table 14: Official-split results over five paired seeds. Performance values are mean ± standard deviation. Lower is better for Brier score and expected calibration error (ECE); higher is better otherwise.

(a) Performance across five paired seeds.

Metric	Sup	Full CTP	Unlabeled-only
Accuracy	0.7840 ± 0.0072	0.8233 ± 0.0058	0.8267 ± 0.0097
AUROC	0.8777 ± 0.0031	0.8996 ± 0.0039	0.9007 ± 0.0032
AUPRC	0.7925 ± 0.0118	0.8403 ± 0.0059	0.8405 ± 0.0034
MCC	0.5555 ± 0.0143	0.6209 ± 0.0123	0.6330 ± 0.0209
Bal. accuracy	0.7863 ± 0.0072	0.8134 ± 0.0063	0.8220 ± 0.0109
Brier	0.1556 ± 0.0026	0.1361 ± 0.0043	0.1341 ± 0.0037
ECE	0.1253 ± 0.0098	0.1029 ± 0.0090	0.1024 ± 0.0115

(b) Mean paired differences.

Metric	Full − Sup	Unlabeled − Sup	Unlabeled − Full
AUROC	$+0.0219$	$+0.0230$	$+0.0011$
Accuracy	—	$+0.0427$	$+0.0033$
AUPRC	—	$+0.0480$	$+0.0002$
MCC	—	$+0.0775$	$+0.0121$
Bal. accuracy	—	$+0.0358$	$+0.0087$
Brier	—	-0.0215	-0.0019
ECE	—	-0.0229	-0.0005

Table 15: Counterfactual-sampling analysis on the official split (three seeds). Mixed replacement draws target-only, E3-only, and joint replacements.

Sampling policy	Accuracy	AUROC	AUPRC	MCC	ECE
E3 only	0.8300	0.8971	0.8341	0.6421	0.1065
Frequency matched	0.8200	0.8932	0.8145	0.6201	0.1098
Full mixed	0.8378	0.9054	0.8492	0.6523	0.0918
Hard mixed	0.8267	0.8977	0.8240	0.6345	0.1008
Joint target–E3 only	**0.8444**	0.9086	**0.8499**	**0.6677**	0.1030
Target only	0.8289	**0.9088**	0.8396	0.6377	0.0954

Table 16: Pretraining sensitivity on the official split (three seeds per setting). The default setting is $\tau_{\text{tuple}} = \tau_{\text{mol}} = 0.1$ and $\lambda_{\text{mol}} = 0.1$.

Parameter	Values	Corresponding AUROC	Range
τ_{tuple}	0.05 / 0.10 / 0.20	0.9017 / 0.8991 / 0.9010	0.0026
τ_{mol}	0.05 / 0.10 / 0.20	0.9011 / 0.8991 / 0.9001	0.0020
λ_{mol}	0.05 / 0.10 / 0.20 / 0.50	0.8994 / 0.8991 / 0.9041 / 0.9017	0.0050

ESM2-650M integration. For the frozen variants, mean-pooled ESM2-650M embeddings supply the target and E3 representations and remain fixed while the DegradeQuery tuple predictor is trained. The supervised fine-tuning reference unfreezes the final ESM2 layer and its final layer normalization. It uses bfloat16 (BF16) precision, a physical batch size of 2 with 16 gradient-accumulation steps, a maximum sequence length of 512, AdamW with learning rate 10^{-5}, and up to 10 epochs. It has 21.45M trainable ESM2 parameters. All variants retain the degrader input and role-conditioned tuple fusion.

Table 17: ESM2-650M comparison on the official split (three seeds, mean ± standard deviation).

Tuple predictor	Trainable ESM2 parameters	AUROC	Accuracy
DegradeQuery-Sup, frozen ESM2	0	0.8863±0.0010	0.7933±0.0067
DegradeQuery-Sup, last-layer FT	21.45M	0.8907±0.0034	0.8144±0.0150
DegradeQuery-CTP, frozen ESM2	0	**0.9032±0.0020**	**0.8322±0.0107**

Repeated target–E3 group holdouts. Each split holds out complete target–E3 groups until approximately 20% of labeled rows are assigned to test; those groups are excluded from both pretraining and supervised training. Table 18 shows the mean performance and variation across ten independently constructed splits.

Table 18: Repeated target–E3 group holdout results over ten splits. Differences are CTP − Sup. Negative differences favor CTP for Brier score and ECE.

Metric	Sup mean ± SD	CTP mean ± SD	Mean paired difference
Accuracy	0.6777±0.0325	0.6717±0.0601	−0.0060
AUROC	0.6769±0.0906	0.7064±0.1027	+0.0295
AUPRC	0.5964±0.0915	0.6110±0.1191	+0.0146
MCC	0.2810±0.0877	0.2947±0.1361	+0.0138
Balanced accuracy	0.6235±0.0461	0.6362±0.0663	+0.0127
Brier score	0.2826±0.0378	0.2725±0.0507	−0.0101
ECE	0.2727±0.0351	0.2498±0.0504	−0.0228

The paired 95% confidence intervals for accuracy, AUROC, AUPRC, and MCC are $[-0.0400, 0.0280]$, $[-0.0219, 0.0810]$, $[-0.0561, 0.0852]$, and $[-0.0580, 0.0855]$, respectively. The intervals for balanced accuracy, Brier score, and ECE are $[-0.0239, 0.0494]$, $[-0.0432, 0.0230]$, and $[-0.0595, 0.0138]$, respectively.

The mean AUROC difference is positive, while the split-level interval shows that transfer depends on held-out context composition. In the original grouped split, the largest held-out target group (P10275) contains 180 of 300 examples and the largest recruiter group (Q96SW2) contains 172; smaller groups such as O43924 and P00533 contain 10 examples each and have positive rates of 0.10. These diagnostics document concentrated and imbalanced biological coverage in the original grouped split.

Relational-null control. We shuffle target–E3 pair assignments during pretraining while preserving the marginal record structure to test whether the exact observed pairing is uniquely responsible for the gain. Table 19 shows no observed-tuple advantage over this control. This result supports the effectiveness of tuple-conditioned pretraining while leaving the unique contribution of observed pair semantics unresolved.

Table 19: Relational-null control over three paired seeds. The difference is shuffled-pair − observed-tuple CTP.

Metric	Observed tuples	Shuffled target–E3 pairs	Mean paired difference
Accuracy	0.8256±0.0069	0.8411±0.0051	+0.0156
AUROC	0.8991±0.0050	0.9021±0.0082	+0.0029
AUPRC	0.8386±0.0071	0.8336±0.0052	−0.0050
MCC	0.6251±0.0154	0.6693±0.0111	+0.0442
ECE	0.1091±0.0036	0.0903±0.0059	−0.0188

The paired 95% confidence intervals for accuracy, AUROC, AUPRC, MCC, and ECE are $[-0.0135, 0.0446]$, $[-0.0258, 0.0317]$, $[-0.0312, 0.0212]$, $[-0.0165, 0.1049]$, and $[-0.0358, -0.0019]$, respectively.

Calibration control. Temperature scaling is fitted exclusively on train-side calibration data; Table 20 reports ECE and negative log-likelihood (NLL). Scaling improves both models and narrows the calibration gap; these results do not support an intrinsic calibration advantage for CTP.

Table 20: Post-hoc temperature-scaling control over three seeds.

Method	Raw ECE	Scaled ECE	Raw NLL	Scaled NLL
Sup	0.0703	0.0626	0.4300	0.4237
CTP	0.0994	0.0772	0.4671	0.4357

External-cohort and endpoint audit. The available PROTAC-Bench release incorporates PROTAC-DB 3.0 and shares 2,461 canonical SMILES and 156 targets with the PROTAC-8K resources, so it does not provide a source-independent external benchmark. Its released benchmark table contains no sample-level DC_{50} or $D_{\max}$ values, preventing matched re-binarization under alternative activity thresholds. The overlap, endpoint conventions, label distribution, and split construction would be confounded in a direct numerical comparison; a deduplicated, endpoint-harmonized benchmark with a predeclared split is required for external validation.

The released benchmark also lacks disease annotations and contains sparse cell-line metadata, which precludes a statistically reliable disease-stratified analysis. Teacher-guided activity targets may provide an additional signal, while introducing a different supervision source and assumptions about teacher reliability; the experiments here isolate the information supplied by recorded tuple contexts themselves.

A.6 Few-Shot Adaptation Protocol

Target-wise few-shot adaptation evaluates whether tuple pretraining provides a better initialization for target-conditioned degradation prediction under limited target-specific supervision. For each held-out target, the support set contains K positive and K negative labeled examples. The query set consists of the remaining labeled rows for that target and is used only for evaluation. The $K = 2$, $K = 4$, and $K = 8$ settings are treated as separate adaptation protocols because the query set changes with K. The base model excludes the held-out targets from pretraining and supervised training. For each target, only the classifier head is adapted for 20 steps with learning rate 10^{-4}; the encoder remains fixed. Metrics are averaged across target episodes within each run and then across the three seeds for each method and K.

Table 21: Target-wise few-shot adaptation protocol.

Setting	Positive support	Negative support	Query rows
$K = 2$	2 per held-out target	2 per held-out target	Remaining labeled rows for the held-out target; evaluation only
$K = 4$	4 per held-out target	4 per held-out target	Remaining labeled rows for the held-out target; evaluation only
$K = 8$	8 per held-out target	8 per held-out target	Remaining labeled rows for the held-out target; evaluation only

A.7 Evaluation Metrics

AUROC is the primary ranking metric. Accuracy is reported for compatibility with the PROTAC-8K benchmark, and AUPRC, MCC, balanced accuracy, and calibration-related metrics are used as secondary diagnostics.

For threshold-dependent metrics, predictions are binarized using the fixed threshold $\delta = 0.5$:

$$\hat{y}_i = \mathbb{I}[\hat{p}_i \geq 0.5]. \tag{16}$$

Let n denote the number of evaluated examples. TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively.

Accuracy is computed as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{17}$$

Balanced accuracy is computed as the average of sensitivity and specificity:

$$\text{BalAcc} = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{TN} + \text{FP}}\right). \tag{18}$$

MCC is computed as

$$\text{MCC} = \frac{\text{TPTN} - \text{FPFN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}. \tag{19}$$

The Brier score is

$$\text{Brier} = \frac{1}{n}\sum_{i=1}^{n}(\hat{p}_i - y_i)^2. \tag{20}$$

The negative log-likelihood is

$$\text{NLL} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log \hat{p}_i + (1 - y_i)\log(1 - \hat{p}_i)\right]. \tag{21}$$

Expected calibration error is computed by partitioning positive-class probabilities into $B = 10$ equal-width bins $\{\mathcal{I}_b\}_{b=1}^{B}$. Let $\bar{y}_b$ be the positive-label fraction and $\bar{p}_b$ the mean predicted probability in bin b:

$$\text{ECE} = \sum_{b:\,|\mathcal{I}_b|>0} \frac{|\mathcal{I}_b|}{n}\left|\bar{y}_b - \bar{p}_b\right|. \tag{22}$$

A.8 Result Reporting

The appendix reports the complete supplementary analyses summarized in the main text, including their statistical units, uncertainty estimates, and protocol boundaries. Official-split reliability and unlabeled-only comparisons use seeds 1–5. Component, ESM2, scaffold-holdout, and few-shot comparisons use seeds 1–3; target–E3 group holdout uses ten independently constructed splits. Standard deviations are sample standard deviations. Paired intervals use Student's t distribution over the per-seed or per-split metric differences. Paired 95% confidence intervals are reported in the text for the corresponding comparisons. Tables 14–20 distinguish seed replication, independently constructed group holdouts, relational-null testing, and post-hoc calibration. All results use the same label preprocessing and metric implementation. Threshold-dependent metrics use the fixed threshold of 0.5, including after post-hoc temperature scaling.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state that DegradeQuery treats PROTAC degradation as a molecule–target–E3 tuple-conditioned prediction problem and uses counterfactual tuple pretraining without pseudo-labels, teacher models, distillation, or ensembles. These claims are directly supported by the method description in Section 3 and the experimental results in Section 4, including the official PROTAC-8K benchmark, ablations, holdout evaluations, and few-shot adaptation experiments.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses limitations in the conclusion and technical appendices, including label sparsity in PROTAC-8K, limited assay metadata in the normalized benchmark, dependence on public benchmark quality, and the scope of retrospective degradation prediction. The claims are restricted to the reported benchmark and generalization settings.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs.
   - The authors should reflect on the factors that influence the performance of the approach.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results, theorems, or proofs. The equations in Section 3 define the model, pretraining objective, and fine-tuning loss rather than stating theoretical guarantees.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.

- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Sections 3 and 4 describe the model components, pretraining and fine-tuning objectives, dataset construction, benchmark splits, metrics, threshold policy, and evaluation settings needed to reproduce the main claims. The appendix further documents the split protocol, leakage controls, data-processing rules, and configuration details for the reported experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways, including detailed instructions, access to a hosted model, release of a model checkpoint, or other appropriate means.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility.

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

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

   Answer: [Yes]

   Justification: The public DegradeQuery repository provides the implementation, preprocessing scripts, configuration files, fixed split files, and reproduction instructions. It describes how to obtain and prepare PROTAC-8K and run the main experiments.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - While release of code and data is encouraged, [No] is an acceptable answer when release is not possible or not yet provided.
   - The instructions should contain the exact command and environment needed to reproduce the results.
   - The authors should provide instructions on data access and preparation.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines, or state which ones are omitted and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions if applicable.

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: Section 4 and the appendix specify the official PROTAC-8K split, target–E3 group holdouts, scaffold holdout, and target-wise few-shot adaptation, as well as metrics and fixed-threshold evaluation. The appendix provides model configuration, optimization, pretraining objectives, negative sampling policy, fine-tuning loss, and model-selection details.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: The paper reports means and standard deviations over repeated runs and paired confidence intervals for the main comparisons. It specifies the statistical units, including random seeds for official-split and ESM2 analyses and independently constructed target–E3 group holdouts.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests.
- The factors of variability that the error bars are capturing should be clearly stated.
- The method for calculating the error bars should be explained.
- The assumptions made should be given.
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- If error bars are reported in tables or plots, the authors should explain how they were calculated.

8. **Experiments compute resources**

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

Answer: [No]

Justification: The appendix provides training configurations and ESM2 fine-tuning settings. It does not provide a complete accounting of hardware resources, per-run execution times, or total compute.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers, CPU or GPU, internal cluster or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each experimental run as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper.

9. **Code of ethics**

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

Answer: [Yes]

Justification: The work conforms to the NeurIPS Code of Ethics. It uses public scientific data and retrospective computational evaluation and does not involve human subjects or private personal data.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity if applicable.

10. **Broader impacts**

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

Answer: [No]

Justification: The paper describes potential benefits for data-efficient degradation prediction and the limits of retrospective evaluation. It does not include a dedicated discussion of both positive and negative societal impacts.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include malicious or unintended uses, fairness considerations, privacy considerations, and security considerations.
- The authors should consider possible harms that could arise when the technology is used as intended, when it gives incorrect results, and from misuse.

11. **Safeguards**

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

Answer: [N/A]

Justification: The released assets are task-specific code, experiment configurations, processed benchmark splits, and documentation for retrospective PROTAC degradation prediction.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model.
- Datasets that have been scraped from the Internet could pose safety risks and should include appropriate safeguards where needed.

12. **Licenses for existing assets**

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

Answer: [No]

Justification: The paper credits the benchmark and external methods, and the released implementation includes an MIT license. The paper does not include a complete inventory of licenses and terms of use for all third-party assets.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license should be included for each asset.
- For scraped data from a particular source, the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The released assets include code, experiment configurations, processed split files, and documentation for reproducing the main benchmark and generalization results. The accompanying documentation describes the intended use, data preprocessing, training/evaluation commands, limitations, and license for the released materials.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets if applicable.

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects. All experiments are computational evaluations on public or processed scientific datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board approvals**

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

    Answer: [N/A]

    Justification: The paper does not involve human subjects, participant data, or crowdsourced annotation. Therefore, IRB approval or equivalent human-subjects review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval or equivalent may be required for any human-subjects research.
- We expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity, if applicable.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method does not use LLMs as an important, original, or non-standard research component. Any LLM use, if present, was limited to writing, editing, or formatting assistance and did not affect the methodology, experiments, scientific rigor, or originality of the work.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - If the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.
    - Please refer to the LLM policy in the NeurIPS handbook for what should or should not be described.