UNIFYING PROTEIN DOMAIN PARSING AND CLASSIFICATION AT SCALE

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein domains are modular units of protein structure that play central roles in protein folding, function, and evolution. Annotating a protein structure requires answering two coupled questions: which residues form each domain, and what fold each domain adopts. Existing methods largely address these questions separately, with domain parsing and fold classification performed by sequential stages. Such pipelines cannot optimize the two tasks jointly and may propagate parsing errors into subsequent classification. We introduce joint protein domain parsing and fold classification and construct TEDBench-Joint, a large-scale benchmark derived from The Encyclopedia of Domains (TED). We further propose DoTR, an end-to-end SE(3)-invariant model that encodes residue-level 3D structure and uses a transformer decoder to directly predict an unordered set of domains, each represented by a residue mask and a fold label. Experiments on TEDBench-Joint show that DoTR outperforms state-of-the-art domain-parsing methods on parsing metrics and two-stage pipelines on unified parsing and classification metrics. These results demonstrate that domain parsing and fold classification can be effectively learned within a single model, providing a unified approach to protein domain annotation at scale.

1 INTRODUCTION

The concept of protein domains was introduced by Wetlaufer (1973), who described them as structurally compact and relatively separate regions of a protein (Postic et al., 2017). Structural classification systems such as CATH (Orengo et al., 1997) have since organized domains into hierarchical groups according to their structural and evolutionary relationships (Orengo et al., 1997). Annotating a protein structure therefore involves two closely coupled problems: *domain parsing*, which determines which residues constitute each domain, and *fold classification*, which assigns each domain to a structural class.

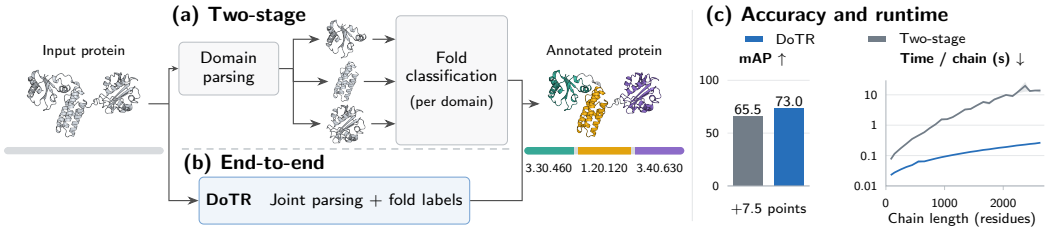

Figure 1: **Unifying protein domain parsing and fold classification.** Panels (a) and (b) contrast the traditional two-stage pipeline and our end-to-end modeling for the same structure. Annotated domains are marked by color and fold labels. (a) The two-stage pipeline sequentially delineates the domains and classifies the topology of the parsed domains. (b) Our end-to-end model, DoTR, predicts the domain locations and their corresponding fold classes jointly. (c) DoTR consistently outperforms the strongest two-stage method while being an order of magnitude faster for long chains.

The rapid growth of predicted structure databases has made domain annotation a proteome- and metagenome-scale problem. The AlphaFold Database (AFDB) contains hundreds of millions of

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.

structures (Varadi et al., 2024), and more recent resources extend predicted structural coverage to over a billion proteins (Candido et al., 2026). The Encyclopedia of Domains (TED) (Lau et al., 2024) addresses this challenge by decomposing AFDB structures into domains and then assigning them to CATH via structure-based alignment (Van Kempen et al., 2024). Building on TED, TED-Bench (Chen et al., 2026a) introduced the first AlphaFold-scale non-redundant fold classification benchmark. However, TEDBench deliberately reduces each protein to a single classification target: the CATH Topology label of its largest annotated domain. While this formulation enables standardized fold recognition at scale, it does not enable a single model to recover domain boundaries, distinguish multiple domains, and classify every domain within a protein.

More generally, existing approaches continue to separate domain parsing from fold classification. Learning-based methods such as Merizo (Lau et al., 2023) and Chainsaw (Wells et al., 2024) predict domain boundaries without fold identities, whereas supervised fold classifiers operate on pre-segmented domains or protein-level targets (Hou et al., 2018; Chen et al., 2026a). Large-scale annotation systems such as TED (Lau et al., 2024) and DPAM (Zhang et al., 2023) can produce both boundaries and classifications, but obtain them through multiple independent stages involving parsing, structural search, and post-processing, with structural search being computationally expensive. Consequently, the two decisions are not learned jointly: fold identity cannot inform domain decomposition, and parsing errors can propagate to downstream classification.

We ask whether protein domain annotation can instead be learned as a single structured prediction problem: given a protein, can a single model directly identify *all* of its constituent domains and assign a fold class to each one? Despite substantial progress on domain parsing and fold classification individually, existing benchmarks isolate these tasks. Parsing benchmarks evaluate domain decomposition without domain topology classification, while classification benchmarks evaluate fold recognition without domain localization. To our knowledge, no existing large-scale benchmark jointly evaluates domain parsing and fold classification from complete protein structures.

We therefore introduce TEDBench-Joint, a first large-scale benchmark that formulates protein domain parsing and classification as a joint domain annotation task (Figure 1). Each example consists of a protein together with its constituent domain instances and their CATH Topology labels, requiring models to jointly determine *where* each domain is and *what fold* it adopts. We further introduce DoTR (Domain TRansformer), an end-to-end SE(3)-invariant model that maps a protein directly to an unordered set of domains, each represented by a residue-level membership mask and a fold label. Experiments on TEDBench-Joint show that DoTR outperforms existing domain-parsing methods on parsing metrics and two-stage pipelines on unified parsing and classification metrics (*unified metrics*), providing a unified approach to protein domain annotation at scale.

2 Related Work

Our work unifies protein domain parsing and fold classification through end-to-end set prediction. We review related work in these areas and in set prediction models most related to DoTR.

For the sake of space saving, we provide a detailed review in Appendix A. Existing learning methods and benchmarks address either domain parsing or fold classification, but not the joint recovery and classification of all domain instances. We make the following contributions: (1) *we construct TEDBench-Joint, to our knowledge the first large-scale benchmark for unified protein domain parsing and fold classification, using structural multi-label splits*; (2) *we introduce DoTR, an end-to-end SE(3)-invariant model that predicts a set of (fold class, residue mask) pairs in a single network forward pass*; (3) *we show that DoTR consistently outperforms existing domain-parsing methods on parsing metrics and two-stage pipelines on unified metrics*; and (4) *we systematically study architectural choices and failure cases for joint protein domain parsing and fold classification.*

3 Benchmark

TEDBench-Joint is constructed from TED domain annotations (Lau et al., 2024) mapped to identifiable CATH domains (Orengo et al., 1997) across AFDB (Varadi et al., 2024). Our construction pipeline comprises four steps: (1) *filtering* to retain reliable domain annotations and high-confidence predicted structures; (2) *structural clustering* to control structural redundancy and define leakage-

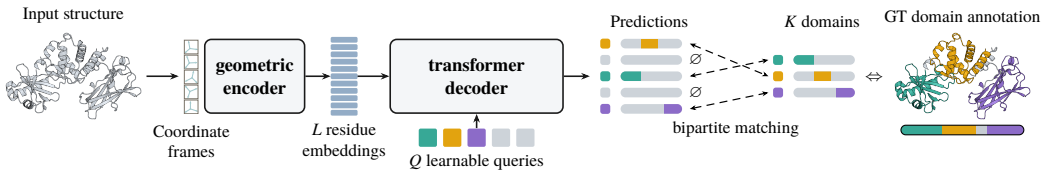

Figure 2: **DoTR architecture.** An input structure is represented as a sequence of coordinate frames fed to a SE(3)-invariant encoder, which maps them to residue embeddings. A two-way transformer decoder refines residue embeddings with a fixed set of learnable query vectors. A class head and a bilinear mask head produce one (class, residue mask) pair per query. During training, bipartite matching assigns predictions to the ground-truth set of domains.

free split units; (3) *multi-label stratification* to preserve fold-label distribution balance across splits; and (4) *cluster sampling* to retain diverse multi-domain proteins and fold-label combinations. We also define parsing metrics for direct comparison with existing domain-parsing methods and unified metrics for jointly evaluating parsing and classification (see Appendix C).

Starting from a set of 324M TED domains in 179M AFDB proteins, we construct TEDBench-Joint with 969K proteins, 1.74M domains, and 1,055 target labels (either T-level or A-level fallback classes). We describe the main construction steps below. Appendix B provides complete details.

Filtering. We first filter domain annotations following the curation principles used to expand the latest CATH release with TED domains (Waman et al., 2025). We retain globular domains containing at least six secondary-structure elements and one CATH label at either the Topology (T) or Homologous Superfamily (H) level. We then retain only proteins for which *all* constituent domains satisfy these criteria and whose predicted structures have high confidence (mean pLDDT $\geq$ 80). Finally, we remove proteins annotated as UniProt fragments (Consortium, 2025), as well as proteins whose Foldseek-cluster representative is a fragment (Barrio-Hernandez et al., 2023). This yields 104M domains in 65M proteins, with a confident structural class label for every retained domain. TED assigns domains by consensus of three parsing methods, and residues claimed by only one of them have an ambiguous assignment. We mark such residues as void, following computer vision benchmarks (Cordts et al., 2016; Kirillov et al., 2019). Overall, 72.7% of residues belong to a domain, 8.6% are non-domain residues (NDRs), and the remaining 18.7% are void.

Structural clustering. We assign each protein to one of the 589K structural clusters defined by Barrio-Hernandez et al. (2023). These clusters serve as the atomic units for data splitting, ensuring that structurally similar proteins are not distributed across training and evaluation sets. Following TEDBench (Chen et al., 2026a), we also merge rare T-level classes that occur in fewer than 10 structural clusters into a coarser A-level label. Unlike TEDBench, where rarity is counted at the protein level, we count occurrences at the cluster level to match the unit of splitting. This reduces the original 1,210 T-groups to 1,055 labels while retaining broad coverage of fold space.

Multi-label stratification. Because a protein may contain multiple domains with different fold labels, TEDBench-Joint requires a multi-label split. We therefore apply iterative stratification (Sechidis et al., 2011) at the *cluster* level rather than the protein level. Each structural cluster is represented by the set of T-level labels appearing in any of its proteins and is assigned in its entirety to the training, validation, or test split in an 80/10/10 ratio. This guarantees that no protein in the validation or test set shares a structural cluster with a training protein, while approximately preserving the marginal distribution of fold labels across splits. The resulting splits are also closely matched in other properties, including residue composition and protein-length distributions (Appendix B). The cluster assignment is fixed and shared across all sampling strategies described below.

Cluster sampling. Although structural clusters prevent leakage across splits, each cluster can still contain substantial redundancy. A standard non-redundant construction would retain only one representative per cluster, e.g. the protein with the highest pLDDT, as in TEDBench. For the present task, however, this strategy disproportionately selects single-domain proteins and retains only one domain-label composition from clusters containing heterogeneous multi-domain proteins. We therefore partition each structural cluster into subgroups according to the multiset of domain labels carried by each protein and retain the highest-pLDDT protein from each subgroup. This strategy preserves the composition of the full set of cluster members: 44.2% of proteins contain more than one do-

main, compared to 44.7% across all cluster members, and only 27.3% when a single representative per cluster is retained. The resulting benchmark contains 969K proteins and 1.74M domains: 775K proteins for training, 97K for validation, and 97K for testing. We additionally provide alternative sampling schemes under the same fixed cluster split to support controlled studies of training-set size, redundancy, and test-set composition (Appendix B).

4 DoTR: Unified Protein Domain Parsing and Classification

We formulate joint protein domain parsing and fold classification as a set prediction problem: given a protein structure, the model jointly identifies domain residues and predicts their corresponding fold classes in a *single network forward pass*. A closely related formulation is widely used in computer vision, where panoptic segmentation models predict sets of (class, pixel mask) pairs (Cheng et al., 2021; 2022), building on the set prediction framework introduced by DETR for object detection (Carion et al., 2020). Our DoTR (Domain TRansformer) generalizes this formulation to proteins. It produces a fixed set of candidate predictions, each represented by a fold class and a residue mask, from which a variable-size set of domains is obtained by discarding no-object predictions. Residues assigned to no predicted domain are treated as non-domain residues (NDRs). Since a residue mask can represent an arbitrary subset of residues without imposing sequence contiguity, this formulation naturally represents discontinuous domains (Wetlaufer, 1973; Lau et al., 2024).

DoTR is illustrated in Figure 2. Backbone coordinates are provided as input to a geometric encoder (Section 4.2), which produces per-residue embeddings. A two-way transformer decoder (Section 4.3) jointly refines a fixed set of learnable queries and the residue embeddings. Each query is then decoded into a (fold class, residue mask) pair. During training, predicted queries are matched to ground-truth domains via Bipartite matching, and the objective combines classification and mask losses (Section 4.4). At inference, the predicted pairs are read out either as potentially overlapping domain masks or as a non-overlapping partition of the chain (Section 4.5).

4.1 Problem formulation

A protein chain consists of L residues with backbone coordinates $\boldsymbol{X} \in \mathbb{R}^{L\times 3\times 3}$ for the atoms $(\mathrm{N}, \mathrm{C}_\alpha, \mathrm{C})$. A *domain* is represented by a pair $(c, \boldsymbol{m})$, where $c \in [C]$ denotes its fold class, given here by the CATH Topology (T-level) label, and $\boldsymbol{m} \in \{0,1\}^L$ is its binary residue mask. The mask defines the domain residue set $\mathbb{V}_{\boldsymbol{m}} = \{i \in [L] : m_i = 1\}$. The mask is *not* required to be a contiguous sequence interval; thus, a discontinuous domain can be represented by a single mask containing multiple sequence segments. A chain contains a set of ground-truth domains $\mathbb{S}^* = \{(c_k^*, \boldsymbol{m}_k^*)\}_{k=1}^K$, where the number of domains K varies across chains and multiple domains may share the same fold class. Residues belonging to no annotated domain are treated as NDRs. The model is trained on a dataset of annotated chains $\{(\boldsymbol{X}_n, \mathbb{S}_n^*)\}_{n=1}^N$. At inference, given an unseen protein chain, the goal is to predict a set of domains $\hat{\mathbb{S}}$ that recovers the annotated domain masks and fold classes.

4.2 Protein encoder

The encoder produces per-residue representations used to delineate domain boundaries and to assign fold classes. 3D structure provides direct information about the spatial organization underlying both domain composition and fold identity. We therefore adopt the residue frame representation and geometric encoder from Chen et al. (2026a), which build on prior work (Ingraham et al., 2019; Hayes et al., 2025). We summarize them here and refer to Chen et al. (2026a) for full details.

Geometric encoder. Each residue is represented by a local coordinate frame describing its position and orientation in 3D space. Specifically, residue i is associated with a rigid transformation $T_i = (\boldsymbol{R}_i, \boldsymbol{t}_i) \in \mathrm{SE}(3)$, where $\boldsymbol{R}_i \in \mathrm{SO}(3)$ is a rotation matrix and $\boldsymbol{t}_i \in \mathbb{R}^3$ is a translation vector. $\boldsymbol{R}_i$ is constructed from an orthonormal basis derived from the backbone atoms $(\mathrm{N}, \mathrm{C}_\alpha, \mathrm{C})$, while $\boldsymbol{t}_i$ is the position of the residue's C_α atom. The encoder takes the sequence of residue frames as input and applies geometric attention blocks followed by standard Transformer blocks. The geometric attention blocks follow ESM3 (Hayes et al., 2025), with minor simplifications (Chen et al., 2026a). By incorporating residue frames into attention, the encoder captures relative geometric information while remaining invariant to global rotations and translations of the input structure.

4.3 TWO-WAY TRANSFORMER DECODER

The decoder maps the per-residue encoder representations to Q candidate domain predictions. As in DETR (Carion et al., 2020), the encoder outputs are linearly projected to decoder dimension d, yielding residue embeddings $\boldsymbol{H} \in \mathbb{R}^{L\times d}$. The decoder jointly updates Q learnable query embeddings $\boldsymbol{Q} \in \mathbb{R}^{Q\times d}$ and the residue embeddings through a stack of two-way attention blocks, following the bidirectional decoder design of SAM (Kirillov et al., 2023; Ravi et al., 2025). GraphDETR (Chen et al., 2026b) adapted this decoder to set prediction on graphs; we adapt the same design to protein chains while retaining positional encodings for residues. Hence, fixed positional encodings $\boldsymbol{E}_q$ and $\boldsymbol{E}_h$ are added at every layer to the queries and residue embeddings, respectively. For residues, we use a one-dimensional version of the normalized sinusoidal encoding used in DETR (Carion et al., 2020). After the final decoder block, the model produces updated query embeddings $\boldsymbol{Q}_{\mathrm{out}}$ and refined residue embeddings $\boldsymbol{H}_{\mathrm{ref}}$. The full details are provided in Appendix D.1.

Each output query $\boldsymbol{q}_j^{\mathrm{out}} \in \mathbb{R}^d$ in $\boldsymbol{Q}_{\mathrm{out}}$ is decoded into a fold-class prediction and a residue mask. The classification head predicts logits over the C fold classes and an additional no-object class $\varnothing$, while the mask head computes compatibility scores between the query and refined residue embeddings:

$$\hat{\boldsymbol{l}}_j^{\mathrm{cls}} = \boldsymbol{W}_{\mathrm{cls}}\, \boldsymbol{q}_j^{\mathrm{out}} \in \mathbb{R}^{C+1}, \qquad \hat{\boldsymbol{l}}_j^{\mathrm{mask}} = \boldsymbol{H}_{\mathrm{ref}} \boldsymbol{W}_{\mathrm{mask}}\, \boldsymbol{q}_j^{\mathrm{out}} \in \mathbb{R}^{L}, \tag{1}$$

where $\boldsymbol{H}_{\mathrm{ref}} \in \mathbb{R}^{L\times d}$ and $\boldsymbol{W}_{\mathrm{mask}} \in \mathbb{R}^{d\times d}$.

4.4 OPTIMIZATION OBJECTIVE

The number of ground-truth domains K varies across chains, whereas the decoder always produces Q candidate predictions. Following standard set prediction approaches (Carion et al., 2020; Cheng et al., 2021), we choose Q to exceed the maximum number of annotated domains per chain in the dataset. Predictions are unordered, so their correspondence to the ground-truth domains is unknown. We therefore use the Hungarian algorithm for bipartite matching (Kuhn, 1955) to assign each ground-truth domain to a distinct query. Classification and mask losses are applied to the matched predictions, while unmatched queries are trained to predict the no-object class $\varnothing$.

Bipartite matching. Let $\sigma : [K] \to [Q]$ denote an injective assignment of ground-truth domains to prediction slots. We define the pairwise matching cost between query j and domain k as

$$C_{j,k} = -\lambda_{\mathrm{cls}} \operatorname{softmax}\big(\hat{\boldsymbol{l}}_j^{\mathrm{cls}}\big)_{c_k^*} + \lambda_{\mathrm{bce}}\, \ell_{\mathrm{bce}}\big(\hat{\boldsymbol{l}}_j^{\mathrm{mask}}, \boldsymbol{m}_k^*\big) + \lambda_{\mathrm{dice}}\, \ell_{\mathrm{dice}}\big(\hat{\boldsymbol{l}}_j^{\mathrm{mask}}, \boldsymbol{m}_k^*\big), \tag{2}$$

where ℓ_{bce} is the binary cross-entropy loss averaged over residues and ℓ_{dice} is the Dice loss (Milletari et al., 2016). The Dice loss is computed from the mask probabilities $\mathrm{sigmoid}(\hat{\boldsymbol{l}}_j^{\mathrm{mask}})$, while the BCE loss is implemented directly from the mask logits. The optimal assignment is obtained by $\sigma^* = \arg\min_\sigma \sum_{k=1}^{K} C_{\sigma(k),k}$, where the minimum is taken over all injective mappings from $[K]$ to $[Q]$. The matching operation itself is non-differentiable; gradients are propagated only through the losses evaluated after the assignment has been computed. Queries not contained in $\sigma^*([K])$ are assigned the no-object target $\varnothing$.

Loss function. Our training objective is based on Mask2Former (Cheng et al., 2022), with two main modifications. First, we supervise only the output of the final decoder block, without auxiliary losses, following SAM (Kirillov et al., 2023). Second, we compute mask losses over all residues rather than over a sampled subset.

Given the optimal assignment σ^*, the matched mask losses are:

$$\mathcal{L}_{\mathrm{bce}} = \frac{1}{K}\sum_{k=1}^{K} \ell_{\mathrm{bce}}\big(\hat{\boldsymbol{l}}_{\sigma^*(k)}^{\mathrm{mask}}, \boldsymbol{m}_k^*\big), \qquad \mathcal{L}_{\mathrm{dice}} = \frac{1}{K}\sum_{k=1}^{K} \ell_{\mathrm{dice}}\big(\hat{\boldsymbol{l}}_{\sigma^*(k)}^{\mathrm{mask}}, \boldsymbol{m}_k^*\big). \tag{3}$$

The classification loss $\mathcal{L}_{\mathrm{ce}}$ is computed over all Q query slots. Matched queries target the fold class of their assigned ground-truth domain, while unmatched queries target $\varnothing$. To reduce the imbalance between matched and unmatched predictions, the contribution of the no-object class is down-weighted by $w_\varnothing < 1$. The total training objective is:

$$\mathcal{L} = \lambda_{\mathrm{cls}}\mathcal{L}_{\mathrm{ce}} + \lambda_{\mathrm{bce}}\mathcal{L}_{\mathrm{bce}} + \lambda_{\mathrm{dice}}\mathcal{L}_{\mathrm{dice}}. \tag{4}$$

Table 1: **Benchmark results on TEDBench-Joint** for DoTR, empirical upper bound, and domain parsing baselines. Chainsaw is retrained on the same training set as DoTR, while methods with * denote the official released checkpoints evaluated directly on our test set. "–" indicates that the field is not applicable. Higher is better for all performance metrics.

		Parsing metrics				Unified metrics			
Model	**Size**	**IoU**	**FracParsed**	**Boundary**	**NDO**	**mAP**	**AP$_{50}$**	**AP$_{75}$**	**PQ**
supervised from scratch									
DoTR-N	2.4M	93.7	92.8	81.7	93.8	65.2	69.4	67.8	71.9
DoTR-T	9.7M	94.5	93.7	83.8	94.6	70.5	74.5	73.0	75.1
DoTR-S	33.3M	94.6	93.9	84.8	94.9	71.4	75.2	73.7	75.9
DoTR-B	107.3M	94.6	93.9	84.7	94.8	71.7	75.7	74.2	76.5
DoTR-L	343.9M	94.7	94.0	85.1	94.9	71.5	75.5	73.9	75.7
pretrained + fine-tuned									
DoTR-S-FT	33.3M	94.7	94.0	84.8	94.9	72.6	76.6	75.2	76.3
DoTR-B-FT	107.3M	**94.9**	**94.2**	**85.2**	**95.1**	73.0	**76.9**	**75.5**	**77.1**
DoTR-L-FT	343.9M	**94.9**	**94.2**	85.1	95.0	**73.1**	**76.9**	**75.5**	76.9
empirical upper bound									
GT + MiAE-B	–	100.0	100.0	100.0	100.0	83.0	83.0	83.0	86.0
parsing methods									
Chainsaw	0.6M	92.5	89.9	78.6	91.4	–	–	–	–
UniDoc	–	88.9	86.1	68.2	90.2	–	–	–	–
Chainsaw + UniDoc	–	91.0	87.3	75.9	90.0	–	–	–	–
Chainsaw*	0.6M	90.5	87.4	79.5	91.5	–	–	–	–
Merizo*	51.8M	90.4	86.2	78.5	92.0	–	–	–	–

4.5 Inference

At inference, the same query predictions can be decoded either as potentially overlapping domain masks or as a non-overlapping assignment of residues to predicted domains, following Cheng et al. (2021; 2022). Let $\boldsymbol{p}_j = \mathrm{softmax}(\hat{\boldsymbol{l}}_j^{\mathrm{cls}})$ and $\hat{\boldsymbol{m}}_j = \mathrm{sigmoid}(\hat{\boldsymbol{l}}_j^{\mathrm{mask}})$ denote the class and residue-membership probabilities of query j, respectively. A query is considered *active* if its most probable class is not $\varnothing$. For each active query, we define its predicted fold class as $\hat{c}_j = \arg\max_{c\in\{1,\dots,C\}} p_{j,c}$ with confidence $s_j = p_{j,\hat{c}_j}$.

Overlapping domains. Following GraphDETR (Chen et al., 2026b), each active query produces one predicted domain whose residue set is $\hat{\mathbb{V}}_j = \{i \in [L] : \hat{m}_{j,i} > \tau_{\mathrm{mask}}\}$, where $\tau_{\mathrm{mask}} = 0.5$ unless otherwise specified. Masks produced by different queries may overlap. We use this decoding scheme for ranking-based evaluation metrics such as AP.

Non-overlapping domains. For metrics requiring a non-overlapping parsing, we follow the panoptic inference of Mask2Former (Cheng et al., 2022). For each residue i, we assign the residue to the active query maximizing $s_j\hat{m}_{j,i}$, and keep the assignment only if that query claims the residue, i.e., $\hat{m}_{j,i} \geq \tau_{\mathrm{mask}}$. A query that wins fewer than a fraction τ_{ov} of the residues it claims is then discarded as a duplicate. Unlike Mask2Former, which leaves the residues of a discarded query unassigned, we reassign them to the query maximizing $s_j\hat{m}_{j,i}$ among those that kept a segment, an idea we borrow from the panoptic post-processing of DETR (Carion et al., 2020). All residues assigned to the same query form one predicted domain and inherit that query's fold class $\hat{c}_j$. Residues that remain unassigned are designated as NDRs. Thus, NDRs are inferred as unassigned residues rather than predicted by an explicit NDR output class. We use this decoding scheme for metrics that require a non-overlapping domain assignment, including PQ and parsing metrics.

5 Experiments

We use TEDBench-Joint to evaluate DoTR against state-of-the-art (SOTA) baselines. Our experiments mainly address four questions: (1) how well does DoTR parse protein domains compared with existing parsers; (2) how does end-to-end prediction compare with two-stage pipelines; (3) what are the runtime benefits of a unified model; and (4) how challenging is the unified task.

5.1 Experimental Setup

DoTR. We evaluate DoTR at five encoder sizes under two training protocols: training from scratch and fine-tuning the corresponding MiAE checkpoint from Chen et al. (2026a). All models use the same training recipe, largely following SAM (Kirillov et al., 2023) and DETR (Carion et al., 2020), with only a small number of hyperparameters tuned. During training, each protein is cropped to a randomly sampled window of 256–1024 residues. We select the checkpoint with the highest validation mAP. Void residues are treated as NDRs during training. Full hyperparameters and training details are given in Appendix F.1.

Domain parsing baselines. We compare against Chainsaw (Wells et al., 2024), Merizo (Lau et al., 2023), and UniDoc (Zhu et al., 2023). Chainsaw is trained on TEDBench-Joint using its published protocol. Since Merizo does not release training code, we evaluate its published checkpoint; UniDoc is non-parametric and requires no training. Note that the use of the released Merizo checkpoint introduces some circularity because Merizo predictions contributed to the TED annotation pipeline. We therefore avoid interpreting its results as a fully independent benchmark baseline. For each method, we tune available post-processing choices on the validation split. Following TED, we also evaluate a Chainsaw+UniDoc pipeline in which Chainsaw first removes NDRs before UniDoc is applied. Because none of these methods predicts fold classes, they are evaluated only with parsing metrics. Full details are provided in Appendix F.2.

Two-stage baselines. To construct comparable two-stage systems, we pair each parser with the SOTA fold classifier. Specifically, we fine-tune a pretrained MiAE-B + seq encoder (Chen et al., 2026a) to classify isolated ground-truth domains from the TEDBench-Joint training set. The resulting classifier reaches 92.4% top-1 accuracy and 86.0 macro-F1 over the $1{,}055$ test classes. Each parser first predicts a set of domains, which are then classified independently by MiAE-B + seq, analogous to image segmentation pipelines (Ding et al., 2022; Xu et al., 2023). We also construct an oracle baseline by applying the same classifier to ground-truth domains, providing an *empirical upper bound* on two-stage performance when parsing is perfect (Borji & Iranmanesh, 2019). Appendix F.3 provides full implementation details.

5.2 Benchmark Results

Table 1 summarizes the main results. On the parsing metrics, DoTR consistently outperforms all parsers across model sizes and both training protocols. The largest gain is observed on the boundary distance score, where our best model improves by more than 5 points over the strongest baseline. Boundary localization is also the most difficult aspect of parsing for all methods, yielding the lowest absolute scores among the four parsing metrics. Parameter counts are not directly comparable to Chainsaw, which relies on precomputed features and a separate search procedure. Nevertheless, DoTR-B is nearly $8\times$ faster than Chainsaw at inference despite having substantially more parameters (Figure 6). Similar gains hold when void residues are included in the evaluation (Table 8).

Unlike parsing, which saturates quickly with model size, joint prediction benefits more clearly from increased model capacity and structural pretraining. However, gains beyond 33M parameters are relatively small in the current setting, while MiAE initialization consistently provides moderate improvements across scales. Considering both accuracy and model size, the fine-tuned DoTR-B is considered our strongest overall configuration.

The oracle baseline helps separate errors due to parsing from those due to fold classification. Because it uses ground-truth domain masks, its gap to perfect performance reflects classification errors alone. Our best DoTR model trails this oracle by only 6.1 points at AP_{50}, with the gap widening at stricter overlap thresholds. In contrast, parsing performance is already close to saturation. This suggests that fold classification, rather than coarse domain decomposition, constitutes the main remaining bottleneck, a conclusion further supported by the stratified analysis in Section 5.4.

5.3 DoTR vs. Two-stage Methods

Table 2 compares our best end-to-end model with two-stage pipelines built from state-of-the-art methods for both stages. UniDoc does not produce confidence scores and is therefore excluded from ranking-based metrics. DoTR-B-FT achieves the best result on three of the four joint metrics and is only slightly surpassed in PQ by the pipeline that combines DoTR parsing with MiAE-B + seq

Model	Size	mAP	AP_{50}	AP_{75}	PQ
end-to-end					
DoTR-B-FT	107.3M	**73.0**	**76.9**	**75.5**	77.1
two-stage					
DoTR-B-FT + MiAE-B	209.9M	69.6	73.3	71.9	**77.9**
Chainsaw + MiAE-B	103.3M	65.5	72.7	67.8	72.9
Merizo* + MiAE-B	154.5M	57.8	68.3	60.1	68.1
UniDoc + MiAE-B	—	—	—	—	69.0
Chainsaw + UniDoc + MiAE-B	103.3M	—	—	—	70.6
empirical upper bound					
GT + MiAE-B	—	83.0	83.0	83.0	86.0

Table 2: **End-to-end DoTR vs. two-stage methods.** Two-stage methods combine existing parsing methods with the SOTA fold classifier MiAE-B (short for the fine-tuned MiAE-B + seq model). Higher is better for all performance metrics.

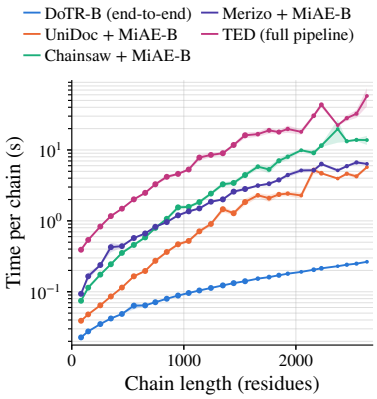

Figure 3: **Runtime comparison.** Time per chain from structure file to full domain annotation.

classification. As this pipeline uses the same predicted domain masks as DoTR, the comparison isolates the classification stage. MiAE-B + seq assigns slightly more accurate labels to individual domains, which benefits PQ, whereas DoTR produces better ranking quality, resulting in higher AP. One plausible explanation is that the standalone classifier has greater classification capacity, particularly for rare classes, but it also requires a second network.

Pipelines built from external parsers perform consistently worse than DoTR, and their ranking largely follows the parsing quality of the first stage in Table 1. The degradation is especially pronounced at stricter overlap thresholds: Chainsaw and Merizo drop by 4.9 and 8.2 points, respectively, from AP_{50} to AP_{75}, compared with only 1.4 points for DoTR. The same pattern is visible relative to the oracle classifier: replacing ground-truth domains with DoTR predictions reduces performance by 13.4 mAP points (83.0 reduced to 69.6), while replacing them with external parsers incurs a further drop. Figure 5 illustrates two examples where DoTR successfully localizes the domains while the representative two-stage method incorrectly merges or splits domains. These results illustrate the error propagation inherent to two-stage pipelines and show that joint optimization is particularly beneficial when accurate localization and confidence ranking are both required. Additional illustrations are shown in Figure 7.

Inference runtime comparison. Following Lau et al. (2023), we measure inference time as a function of protein length. Proteins are grouped into 100-residue bins up to 2,661 residues, with at most 30 proteins sampled per bin. Runtime is measured from structure loading to the final domain annotations, following the benchmark protocol of Wells et al. (2024). Each baseline is run using the official released code, configuration, and environment. We evaluate DoTR-B using one H100 GPU, eight CPU threads, and batch size one for all methods. Two-stage runtime is computed as the sum of parsing and classification times, and TED's as the sum of its pipeline stages. Details of our setup are provided in Appendix G.

Figure 3 shows that DoTR-B is faster than all the baselines across the full range of protein lengths. Averaged over the benchmark length distribution, DoTR-B requires 0.044 s per protein, compared with 0.122 s for UniDoc, 0.345 s for Chainsaw, 0.398 s for Merizo, and 1.437 s for the full TED pipeline, corresponding to speedups of $2.8\times$ to $33\times$. Similar gains are observed against the parsing methods alone (Figure 6). The speed advantage is consistent with avoiding the iterative post-processing and domain-assignment procedures used by Merizo and Chainsaw, together with implementation-level differences. TED compounds these costs by introducing further post-processing and uses structural alignment for classification.

5.4 Stratified and Ablation Studies

We stratify the test set along six axes and report one unified metric (mAP) and one parsing metric (IoU). Three axes describe the protein chain (number of domains, chain length, and mean pLDDT), two describe individual domains (continuity and size), and one describes the target class (training

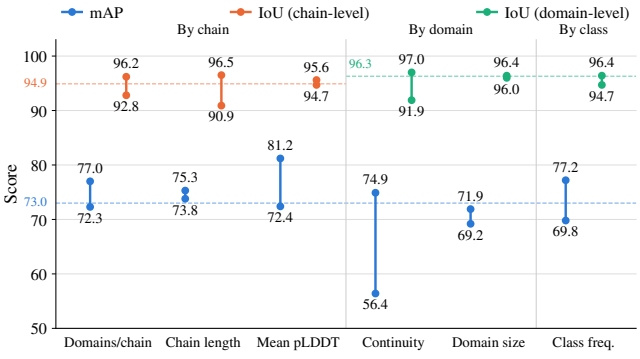

Figure 4: **Stratification results.** Bars span worst-to-best buckets. Dashed lines mark overall scores. mAP is defined identically throughout. IoU is length-weighted by chain for chain axes and per-domain otherwise.

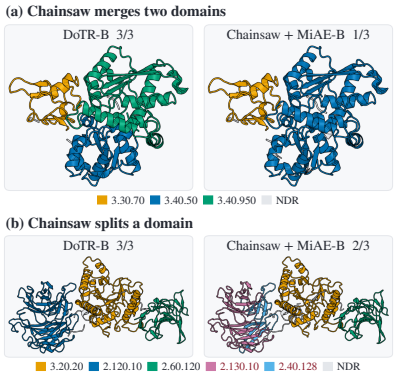

Figure 5: **DoTR vs two-stage methods.** Chainsaw incorrectly merges (a) or splits (b) domains.

frequency, binned, following Gupta et al. (2019)). For mAP, detections are pooled by class and each stratum determines which ground-truth domains contribute to the evaluation, following the protocol of Lin et al. (2014). IoU is averaged over proteins or domains depending on the stratification variable. Figure 4 shows substantially greater variation in mAP than in IoU, which remains between 90.9 and 97.0 across all strata. Discontinuous domains are the clearest failure mode, with an 18.5-point drop in mAP relative to continuous domains. This is consistent with previous observations that discontinuity is a major challenge for domain parsing (Durham et al., 2025), and our results show that the difficulty persists at large scale in the joint task.

Structure quality is the second strongest factor, with mAP increasing from 72.4 to 81.2 between the lowest and highest pLDDT bins, while rare classes trail frequent classes by 7.4 mAP. The corresponding IoU ranges are only 0.9 and 1.7 points, respectively, possibly suggesting that these effects arise primarily from fold classification rather than parsing. Chain length shows the opposite pattern: IoU drops by 5.6 points for longer proteins, while mAP varies by only 1.5 points, indicating robustness of the joint predictions despite more difficult parsing. Domain size shows no consistent trend. Overall, the stratified results identify fold classification and discontinuous domains as the main sources of remaining error.

Appendix E.2 reports ablations of the main architectural and training choices in DoTR. We vary the number of decoder queries, positional encodings applied to decoder keys, decoder depth, training crop size, loss components, and sequence information in the encoder. The default configuration in Table 13 performs best for all factors except the number of queries and the removal of Dice loss. This suggests that the reported model could be further improved through more extensive optimization.

6 Conclusion

We introduced a unified formulation of protein domain parsing and fold classification, replacing the conventional separation between domain delineation and fold assignment with a single set prediction task. To support this setting, we constructed TEDBench-Joint, the first AlphaFold-scale benchmark for standardized evaluation of this unified problem. We further proposed DoTR, an end-to-end model that directly predicts a set of (residue mask, fold class) pairs from a complete protein structure.

Experiments on TEDBench-Joint show that DoTR outperforms existing domain-parsing methods and two-stage pipelines for the unified task, while substantially reducing inference time. Our analyses further identify key architectural choices and remaining challenges, particularly handling of discontinuous domains. Overall, TEDBench-Joint and DoTR establish a large-scale framework for studying end-to-end protein domain annotation and provide a foundation for future advances in unified structural categorization at scale.

Limitations. We provide a discussion on limitations in Appendix H.

AI USE STATEMENT

In this work, we used generative AI tools for finding related work, assisting in writing, modifying figures, assisting in editing software code, and brainstorming. We have not used generative AI tools for developing theoretical models or conceptual frameworks, generating synthetic datasets, or formulating mathematical claims. We have reviewed all AI-assisted work. 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 develops methods for unified protein domain parsing and fold classification and does not involve human subjects, animals, or personal data. We foresee no harm to individuals, groups, or the environment.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, we describe our benchmark construction in Section 3 and Appendix B, including filtering steps, main properties, and statistics. We also describe our model in detail in Section 4 and Appendix D. Finally, we report hyperparameter configurations and selection procedures for DoTR in Appendix F.1, and for the baselines in Appendices F.2 and F.3.

REFERENCES

Inigo Barrio-Hernandez, Jingi Yeo, Jürgen Jänes, Milot Mirdita, Cameron LM Gilchrist, Tanita Wein, Mihaly Varadi, Sameer Velankar, Pedro Beltrao, and Martin Steinegger. Clustering predicted structures at the scale of the known protein universe. *Nature*, 622(7983):637–645, 2023.

Ali Borji and Seyed Mehdi Iranmanesh. Empirical upper bound in object detection and more. *CoRR*, abs/1911.12451, 2019. URL http://arxiv.org/abs/1911.12451.

Salvatore Candido, Thomas Hayes, Alexander Derry, Roshan Rao, Zeming Lin, Robert Verkuil, Bryan Z Wu, Jin Sub Lee, Elise S Bruguera, Jehan A Keval, et al. Language modeling materializes a world model of protein biology. *bioRxiv*, pp. 2026–06, 2026.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Dexiong Chen, Andrei Manolache, Mathias Niepert, and Karsten Borgwardt. Protein fold classification at scale: Benchmarking and pretraining. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026a.

Dexiong Chen, Till Hendrik Schulz, and Karsten Borgwardt. End-to-end subgraph detection with graphdetr, 2026b. URL https://arxiv.org/abs/2606.06364.

Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in neural information processing systems*, 34:17864–17875, 2021.

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1280–1289. IEEE, 2022.

Hua Cheng, R Dustin Schaeffer, Yuxing Liao, Lisa N Kinch, Jimin Pei, Shuoyong Shi, Bong-Hyun Kim, and Nick V Grishin. Ecod: an evolutionary classification of protein domains. *PLoS computational biology*, 10(12):e1003926, 2014.

The UniProt Consortium. Uniprot: the universal protein knowledgebase in 2025. *Nucleic Acids Research*, 53(D1):D609–D617, 01 2025. ISSN 1362-4962. doi: 10.1093/nar/gkae1010. URL https://doi.org/10.1093/nar/gkae1010.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Zheng Ding, Jieke Wang, and Zhuowen Tu. Open-vocabulary universal image segmentation with maskclip. *arXiv preprint arXiv:2208.08984*, 2022.

Jesse Durham, Jing Zhang, Richard D Schaeffer, and Qian Cong. Dpam-ai: a domain parser for alphafold models powered by artificial intelligence. *Bioinformatics*, 41(1):btae740, 2025.

Raphael R Eguchi and Po-Ssu Huang. Multi-scale structural analysis of proteins by deep semantic segmentation. *Bioinformatics*, 36(6):1740–1749, 2020.

Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5351–5359. IEEE, 2019.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025.

Liisa Holm and Chris Sander. Protein structure comparison by alignment of distance matrices. *Journal of molecular biology*, 233(1):123–138, 1993.

Liisa Holm and Chris Sander. Parser for protein folding units. *Proteins: Structure, Function, and Bioinformatics*, 19(3):256–268, 1994.

Jie Hou, Badri Adhikari, and Jianlin Cheng. Deepsf: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303, 2018.

John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. *Advances in neural information processing systems*, 32, 2019.

Arian Jamasb, Alex Morehead, Chaitanya Joshi, Zuobai Zhang, Kieran Didi, Simon Mathis, Charles Harris, Jian Tang, Jianlin Cheng, Pietro Lio, et al. Evaluating representation learning on the protein structure universe. In *International Conference on Learning Representations*, volume 2024, pp. 25597–25627, 2024.

Long Jin, Zeyu Chen, and Zhuowen Tu. Object detection free instance segmentation with labeling transformations. *CoRR*, abs/1611.08991, 2016. URL `http://arxiv.org/abs/1611.08991`.

Shaun M Kandathil, Andy M Lau, Daniel WA Buchan, and David T Jones. Foldclass and merizo-search: scalable structural similarity search for single-and multi-domain proteins using geometric learning. *Bioinformatics*, 41(5):btaf277, 2025.

Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 9396–9405. Ieee, 2019.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Andy M Lau, Shaun M Kandathil, and David T Jones. Merizo: a rapid and accurate protein domain segmentation method using invariant point attention. *Nature Communications*, 14(1):8445, 2023.

Andy M Lau, Nicola Bordin, Shaun M Kandathil, Ian Sillitoe, Vaishali P Waman, Jude Wells, Christine A Orengo, and David T Jones. Exploring structural diversity across the protein universe with the encyclopedia of domains. *Science*, 386(6721):eadq4946, 2024.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

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.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pp. 565–571. Ieee, 2016.

Alexey G Murzin, Steven E Brenner, Tim Hubbard, and Cyrus Chothia. Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4):536–540, 1995.

Christine A Orengo, Alex D Michie, Susan Jones, David T Jones, Mark B Swindells, and Janet M Thornton. Cath–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.

Guillaume Postic, Yassine Ghouzam, Romain Chebrek, and Jean-Christophe Gelly. An ambiguity principle for assigning protein structural domains. *Science advances*, 3(1):e1600552, 2017.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

Oliver C Redfern, Andrew Harrison, Tim Dallman, Frances M G Pearl, and Christine A Orengo. Cathedral: a fast and effective algorithm to predict folds and domain boundaries from multidomain protein structures. *PLoS computational biology*, 3(11):e232, 2007.

Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8216–8223. IEEE, 2023.

Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. On the stratification of multilabel data. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 145–158. Springer, 2011.

Asim S Siddiqui and Geoffrey J Barton. Continuous and discontinuous domains: an algorithm for the automatic generation of reliable protein domain definitions. *Protein Science*, 4(5):872–884, 1995.

Johannes Söding. Protein homology detection by hmm–hmm comparison. *Bioinformatics*, 21(7): 951–960, 2005.

Mark B Swindells. A procedure for detecting structural domains in proteins. *Protein Science*, 4(1): 103–112, 1995.

Chin-Hsien Tai, Woei-Jyh Lee, James J Vincent, and Byungkook Lee. Evaluation of domain prediction in casp6. *Proteins: Structure, Function, and Bioinformatics*, 61(S7):183–192, 2005.

Michael Tress, Jianlin Cheng, Pierre Baldi, Keehyoung Joo, Jinwoo Lee, Joo-Hyun Seo, Jooyoung Lee, David Baker, Dylan Chivian, David Kim, et al. Assessment of predictions submitted for the casp7 domain prediction category. *Proteins: Structure, Function, and Bioinformatics*, 69(S8): 137–151, 2007.

Michel Van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with foldseek. *Nature biotechnology*, 42(2):243–246, 2024.

Mihaly Varadi, Damian Bertoni, Paulyna Magana, Urmila Paramval, Ivanna Pidruchna, Malarvizhi Radhakrishnan, Maxim Tsenkov, Sreenath Nair, Milot Mirdita, Jingi Yeo, et al. Alphafold protein structure database in 2024: providing structure coverage for over 214 million protein sequences. *Nucleic acids research*, 52(D1):D368–D375, 2024.

Vaishali P Waman, Nicola Bordin, Andy Lau, Shaun Kandathil, Jude Wells, David Miller, Sameer Velankar, David T Jones, Ian Sillitoe, and Christine Orengo. Cath v4. 4: major expansion of cath by experimental and predicted structural data. *Nucleic Acids Research*, 53(D1):D348–D355, 2025.

Jude Wells, Alex Hawkins-Hooker, Nicola Bordin, Ian Sillitoe, Brooks Paige, and Christine Orengo. Chainsaw: protein domain segmentation with fully convolutional neural networks. *Bioinformatics*, 40(5):btae296, 2024.

Donald B Wetlaufer. Nucleation, rapid folding, and globular intrachain regions in proteins. *Proceedings of the National Academy of Sciences*, 70(3):697–701, 1973.

Xin Xu, Tianyi Xiong, Zheng Ding, and Zhuowen Tu. Masqclip for open-vocabulary universal image segmentation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 887–898. IEEE, 2023.

Jing Zhang, R Dustin Schaeffer, Jesse Durham, Qian Cong, and Nick V Grishin. Dpam: A domain parser for alphafold models. *Protein Science*, 32(2):e4548, 2023.

Yang Zhang and Jeffrey Skolnick. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research*, 33(7):2302–2309, 2005.

Kun Zhu, Hong Su, Zhenling Peng, and Jianyi Yang. A unified approach to protein domain parsing with inter-residue distance matrix. *Bioinformatics*, 39(2):btad070, 2023.

Appendix

A Detailed Related Work

Our work unifies protein domain parsing and fold classification through end-to-end set prediction. We review related work in these areas and in set prediction models most related to DoTR.

Protein domain parsing. Protein domain parsing aims to partition a protein structure into its constituent domain instances. Classical structure-based methods identify boundaries using geometric or physical criteria, including structural compactness and inter-residue contacts (Swindells, 1995; Siddiqui & Barton, 1995; Holm & Sander, 1994; Postic et al., 2017). UniDoc (Zhu et al., 2023), for example, seeks partitions with strong intra-domain and weak inter-domain interactions. Alignment-based methods instead incorporate homology information to infer boundaries consistent with previously characterized domains (Redfern et al., 2007; Zhang et al., 2023). Recent learning-based approaches include Chainsaw (Wells et al., 2024), which predicts pairwise residue co-membership probabilities with a CNN followed by discrete assignment; DPAM-AI (Durham et al., 2025), which combines learned structural features with similarity-based evidence for AFDB structures (Varadi et al., 2024); and Merizo (Lau et al., 2023), which directly learns domain decomposition using a class-agnostic affinity objective (Jin et al., 2016). These methods delineate domain instances without jointly assigning fold classes. The closest precursor to our formulation is Eguchi & Huang (2020), which predicts one of 38 CATH Architecture classes for each residue. However, this is semantic rather than instance-level segmentation: residues are assigned classes but not grouped into distinct domains. The published model is also restricted to proteins of at most 512 residues (Eguchi & Huang, 2020; Wells et al., 2024).

Protein fold classification. Structural databases such as CATH (Orengo et al., 1997), SCOP (Murzin et al., 1995), and ECOD (Cheng et al., 2014) organize protein domains into hierarchical classifications containing fold-level categories. Traditional assignment methods rely on pairwise structural alignment (Holm & Sander, 1993; Zhang & Skolnick, 2005) or profile-based sequence comparison (Söding, 2005), while recent tools such as Foldseek (Van Kempen et al., 2024) and Foldclass (Kandathil et al., 2025) enable efficient search over large structural databases (Varadi et al., 2024; Lin et al., 2023). Fold classification can also be learned directly as supervised prediction (Hou et al., 2018; Rao et al., 2019; Jamasb et al., 2024; Chen et al., 2026a). TEDBench (Chen et al., 2026a) provides the first large-scale supervised benchmark for this setting, but assigns each protein a single CATH Topology label corresponding to its largest annotated domain. It therefore does not require localization or classification of all domains within a multidomain protein, making it not directly suitable for joint domain parsing and classification.

Set prediction. DETR (Carion et al., 2020) formulates object detection as end-to-end set prediction using learnable queries and Hungarian matching (Kuhn, 1955). MaskFormer and Mask2Former (Cheng et al., 2021; 2022) extend this formulation to mask prediction for universal image segmentation, while related ideas have been applied to 3D point clouds with Mask3D (Schult et al., 2023) and to graphs with GraphDETR (Chen et al., 2026b). Two-way transformer decoders have further improved query-based mask prediction in models such as SAM (Kirillov et al., 2023; Ravi et al., 2025). DoTR adapts this paradigm to protein structures by predicting an unordered set of domain masks and fold labels.

B Benchmark Details

B.1 Filtering

Domain filtering follows the curation principles used to extend CATH with TED domains (Waman et al., 2025). A domain is retained if it is globular, contains at least six secondary-structure elements, and carries a CATH label at the Topology or Homologous Superfamily level. A domain counts as

globular if its packing density is at least 10.333 and its radius of gyration, normalized by domain length, is below 0.356, and secondary-structure elements are counted as helices and strands, excluding turns. A protein is retained only if all of its domains pass these filters, if its AFDB model has a mean pLDDT of at least 80, and if neither the protein nor its cluster representative is annotated as a UniProt fragment. Table 3 reports the number of domains and proteins remaining after each step.

Table 3: Domains and proteins remaining after each filtering step, starting from the TED-100 set.

Step	**Domains**	**Proteins**
TED-100	324,389,697	178,707,533
Globularity	192,968,889	112,218,275
≥ 6 secondary-structure elements	155,017,734	95,938,966
CATH label at T or H level	121,535,084	76,700,219
Mean pLDDT ≥ 80	105,603,478	66,534,011
Fragment removal (filtered set)	103,567,225	65,024,992

B.2 Structural clustering and label merging

Each protein is assigned to its cluster in the AFDB Foldseek clustering of Barrio-Hernandez et al. (2023). The filtered set spans 588,971 clusters. Rare labels are merged before splitting: a T-group that occurs in fewer than ten clusters is replaced by its CATH Architecture, so that every label is supported by at least ten independent clusters. Of the 1,210 original T-groups, 1,033 survive and 177 are merged into 22 Architecture-level buckets, giving 1,055 labels that together cover 1,738,765 domains in the final benchmark.

The label distribution is long-tailed. The most common label holds 231,017 domains (13.3%), the ten largest labels account for 47.2% of all domains, the median label holds 180 domains, and the smallest holds 10. Moreover, 656 labels have at least 100 domains and 184 at least 1,000. By CATH class, 47.1% of domains are α/β, 33.6% mainly β, 19.2% mainly α, and fewer than 0.1% have few secondary structures. The merged Architecture buckets are small, covering 3,982 domains in total.

B.3 Multi-label stratification

Each cluster is represented by the set of labels occurring in any of its proteins and is assigned as a whole by iterative stratification (Sechidis et al., 2011): first training and validation against test in a 90/10 ratio, then training against validation in an 8/9 to 1/9 ratio, over a seeded random permutation of the clusters. Because a cluster's label set records which labels occur but not how many domains carry them, a label whose domains concentrate in one large validation or test cluster can remain unbalanced. A single correction pass therefore exchanges such dominant clusters for training clusters carrying the same label, which preserves the number of clusters per split. Split integrity, i.e., that no cluster appears in two splits, is verified for every protein of the benchmark.

The resulting splits are balanced. Every one of the 1,055 labels occurs in all three splits, the test share of a label's domains deviates from 10% by 1.5 percentage points on average (median 0.9). The largest deviation, 20.8 percentage points, occurs for a label with only 26 domains. Table 4 reports the size of each split.

Table 4: Size of TEDBench-Joint per split. Clusters are the unit of assignment, so no cluster is shared between splits.

	train	**validation**	**test**	**total**
Proteins	775,140	96,863	96,864	968,867
Domains	1,389,948	174,209	174,608	1,738,765
Clusters	471,106	58,989	58,876	588,971
Labels	1,055	1,055	1,055	1,055

B.4 CLUSTER SAMPLING SETTINGS

The split is fixed before sampling, so the same cluster assignment supports several ways of drawing proteins from the clusters. We provide three settings, summarized in Table 5. Sampling all cluster members keeps every protein of the filtered set and is the largest training resource, at the cost of redundancy within clusters. Sampling one protein per cluster is the standard non-redundant construction and keeps the highest-pLDDT protein of each cluster, which biases the set towards single-domain proteins and leaves 45 labels absent from the test split. Sampling one protein per label multiset, the main benchmark, keeps the highest-pLDDT protein of each distinct combination of domain labels within a cluster, which preserves the multi-domain composition of the full set. Because clusters rather than proteins are assigned to the splits, the realized proportions depend on the sampling. The main benchmark and the one-per-cluster setting both land on 80.0/10.0/10.0 by proteins, whereas keeping all cluster members yields 82.2/8.4/9.4, since clusters differ in size.

Table 5: Cluster sampling settings under the same fixed cluster split. The main benchmark used in our experiments is the last row.

Setting	Proteins	Domains	Multi-domain	Discont.	Labels in test
All cluster members	65,024,992	103,567,225	44.7%	21.6%	1,055
One per cluster	588,971	835,542	27.3%	12.5%	1,010
One per label multiset	968,867	1,738,765	44.2%	13.6%	1,055

B.5 BENCHMARK STATISTICS

Table 6 summarizes the composition of the main benchmark. The residue composition is nearly identical across splits, with domain, void and non-domain shares varying by at most 0.3 percentage points.

Table 6: Composition of TEDBench-Joint. The residue composition is given per split.

Property	Value
Protein length (median / mean / 99th pct. / max)	277 / 358 / 1,146 / 2,661
Domain length (median / mean / min / max)	127 / 145 / 33 / 896
Domains per protein (mean / max)	1.8 / 14
Multi-domain proteins	44.2%
Discontinuous domains	13.6%
Residues, domain / void / NDR (train)	72.7% / 18.7% / 8.6%
Residues, domain / void / NDR (validation)	72.5% / 18.9% / 8.6%
Residues, domain / void / NDR (test)	72.7% / 18.7% / 8.6%

B.6 VOID RESIDUES

TED derives its domains from the consensus of three parsing methods and reports, per residue, whether a domain was claimed by all three, by two, or by only one of them. Residues supported by a single method are not part of any domain in our annotation. However, their true assignment is unknown, which is why we mark them as void. They comprise 65.1M residues, or 18.7% of the benchmark.

C METRICS

C.1 METRICS OVERVIEW

We divide evaluation into parsing metrics, which assess whether domain instances are correctly recovered irrespective of fold assignments, and unified metrics, which additionally require correct fold classification. Following Chainsaw (Wells et al., 2024), predicted and ground-truth domains are paired using the one-to-one assignment that maximizes the total intersection over union (IoU),

restricted to pairs of the same fold class for the unified metrics. Following computer vision benchmarks (Kirillov et al., 2019; Lin et al., 2014), void residues are ignored in all tables of the main paper, while we additionally report the results scoring them as NDRs in Appendix E.1.

Parsing metrics. Following prior domain parsing works (Wells et al., 2024; Lau et al., 2023), we report the length-weighted IoU, the fraction of correctly parsed domains (domain-level IoU ≥ 0.8), the domain boundary distance score (Tress et al., 2007), and normalized domain overlap (NDO) (Tai et al., 2005). All parsing metrics are averaged across proteins.

Unified metrics. We report COCO-style mean average precision (Lin et al., 2014) (mAP), averaged over IoU thresholds from 0.50 to 0.95, as well as AP_{50} and AP_{75} at the thresholds of 0.50 and 0.75, respectively. We also report the panoptic quality (PQ) (Kirillov et al., 2019). All unified metrics are computed independently for each fold class and then macro-averaged across labels.

C.2 Detailed Description

All metrics are calculated based on the two inference settings from Section 4.5. Parsing metrics and PQ are calculated on the non-overlapping output, and average precision on the possibly overlapping domains, which carry the confidence scores. Predicted and ground-truth domains are paired as in Chainsaw (Wells et al., 2024), by the one-to-one assignment that maximizes the sum of the intersection over union (IoU) of the paired residue sets, $\mathrm{IoU}(A, B) = |A \cap B|/|A \cup B|$, computed with the Hungarian algorithm (Kuhn, 1955) and restricted to pairs of the same fold class for the unified metrics. Ground-truth domains left without a match have an IoU of zero, and NDRs are never matched.

Parsing metrics. The length-weighted IoU and the fraction of correctly parsed domains are taken from Chainsaw (Wells et al., 2024). The first averages the IoU of each ground-truth domain with its paired prediction, weighting each domain by its length, and the second is the fraction of ground-truth domains whose paired prediction reaches an IoU of at least 0.8. The normalized domain overlap score (NDO) was introduced in (Tai et al., 2005), and it represents the proportion of residues that have been assigned to the correct domain. The boundary distance score was introduced in (Tress et al., 2007), and we use it in the form implemented by Chainsaw. Two conventions of that implementation matter for reading the numbers: every NDR counts as a true boundary position, so a boundary predicted inside a linker is not penalized, and the total is normalized by the larger of the number of predicted and of true boundaries. All parsing metrics are computed per protein and averaged over proteins.

Average precision. We follow the COCO protocol (Lin et al., 2014). For each fold class and each IoU threshold, the predicted domains of that class are pooled over the test set and ranked by their confidence. A prediction whose paired ground-truth domain reaches the threshold counts as a true positive and every other prediction as a false positive. Average precision is the area under the resulting precision–recall curve, obtained by the 101-point interpolation of COCO, and is macro-averaged over the fold classes that occur in the ground truth. We report AP_{50} and AP_{75} at thresholds 0.50 and 0.75, and mAP as the mean over the ten thresholds $0.50, 0.55, \ldots, 0.95$.

Panoptic quality. We follow Kirillov et al. (2019). A paired prediction and ground-truth domain of the same fold class count as a true positive if their IoU exceeds 0.5. Since neither the predicted nor the ground-truth domains overlap, at most one such pair exists per domain, so every matching strategy yields the same true positives. For a given fold class, let TP, FP and FN denote the sets of matched pairs, of unmatched predicted domains and of unmatched ground-truth domains, pooled over the test set. Then:

$$\mathrm{PQ} = \frac{\sum_{(p,g)\in TP} \mathrm{IoU}(p, g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}, \tag{5}$$

which we average over the fold classes. In the terminology of panoptic segmentation, domains are thing classes and there is no stuff class. NDRs enter the intersection over union but are not scored as a class, so our PQ corresponds to $\mathrm{PQ}^{\mathrm{Th}}$.

Void residues. Void residues are removed from the evaluation, following the void rules of panoptic segmentation (Kirillov et al., 2019). For the metrics based on the IoU, they are excluded from

both residue sets before the intersection and the union are formed, and an unmatched prediction whose residues lie predominantly on void is not counted as a false positive. The parsing metrics are positional rather than mask-based, so there the void residues are instead removed from the chain before the score is computed.

D METHODS DETAILS

D.1 TWO-WAY TRANSFORMER DECODER

The decoder maintains the query embeddings $\boldsymbol{Q} \in \mathbb{R}^{Q\times d}$ and the residue embeddings $\boldsymbol{H} \in \mathbb{R}^{L\times d}$, which are jointly updated by a stack of two-way attention blocks. One block works as follows:

$$\begin{aligned} \boldsymbol{Q}^{\ell}_{\text{self}} &= f_{\text{self}}\big(\boldsymbol{Q}^{\ell}+\boldsymbol{E}_q,\ \boldsymbol{Q}^{\ell}+\boldsymbol{E}_q,\ \boldsymbol{Q}^{\ell}\big), & \boldsymbol{Q}^{\ell}_{\text{cross}} &= f_{\text{cross}}\big(\boldsymbol{Q}^{\ell}_{\text{self}}+\boldsymbol{E}_q,\ \boldsymbol{H}^{\ell}+\boldsymbol{E}_h,\ \boldsymbol{H}^{\ell}\big), \\ \boldsymbol{Q}^{\ell+1} &= f_{\text{ffn}}\big(\boldsymbol{Q}^{\ell}_{\text{cross}}\big), & \boldsymbol{H}^{\ell+1} &= f_{\text{rev}}\big(\boldsymbol{H}^{\ell}+\boldsymbol{E}_h,\ \boldsymbol{Q}^{\ell+1}+\boldsymbol{E}_q,\ \boldsymbol{Q}^{\ell+1}\big), \end{aligned} \quad (6)$$

where f_{self}, f_{cross} and f_{rev} are multi-head attention modules with arguments (query, key, value), for query–query, query–residue and residue–query attention, and f_{ffn} is a position-wise feed-forward network. Standard LayerNorm and residual connections are applied to each sub-layer.

Following SAM (Kirillov et al., 2023), the two cross-attentions and the final attention are computed in a reduced internal dimension of $d/2$, whereas the query self-attention uses the full width d. The number of queries, $Q = 20$, exceeds the largest number of annotated domains in any chain of the benchmark, which is 14.

E ADDITIONAL RESULTS

E.1 ROBUSTNESS RESULTS

In this section, we show our main results with bootstrap standard errors for two evaluation settings: void residues ignored and void residues as NDRs. Table 7 and Table 8 show results related to Table 1, while Table 9 and Table 10 show results related to Table 2.

Additionally, these table include the class-agnostic version of DoTR as explained in F.1. DoTR-B-FT-CA performs comparably to our best class-aware models. Moreover, when coupled with the classifier, its two-stage performance matches the performance of our best-to-best models at a much larger number of parameters.

E.2 ABLATION STUDIES

We ablate the key design choices of DoTR. Ablation studies adopt the standard setup from Table 13 for DoTR-S with an exception of crop size, which is fixed to 512 for efficiency. Results are shown in Table 11.

E.3 ADDITIONAL INFERENCE RUNTIME RESULTS

Figure 6 compares the inference runtime of DoTR-B-FT and domain parsing baselines. Our joint model is more efficient than parsing-only methods. The setup is as described in Appendix G.

F EXPERIMENTAL DETAILS

F.1 DOTR TRAINING DETAILS

Training. We train DoTR either from randomly initialized weights or from an encoder pretrained with masked structure reconstruction (Chen et al., 2026a), in which case the decoder and the two heads still start from random initialization. The architecture of each variant is given in Table 12 and the training setting in Table 13. The same setting is used for every model size and for both protocols, so that the two differ only in the encoder initialization. Each batch is cropped to a window drawn from a fixed set of sizes, which exposes the model to lengths of up to 1024 residues, while validation

Table 7: **Benchmark results on TEDBench-Joint.** All void residues are excluded from scoring. $\pm$ denotes $1.96\times$ the bootstrap standard error, from 1000 resamples of the $96{,}864$ test chains drawn with replacement. Point estimates are the values measured on the test set.

		Parsing metrics				Unified metrics			
Model	**Size**	**IoU**	**FracParsed**	**Boundary**	**NDO**	**mAP**	**AP$_{50}$**	**AP$_{75}$**	**PQ**
supervised from scratch									
DoTR-N	2.4M	93.7 ±0.1	92.8 ±0.1	81.7 ±0.1	93.8 ±0.1	65.2 ±0.8	69.4 ±0.8	67.8 ±0.8	71.9 ±0.8
DoTR-T	9.7M	94.5 ±0.1	93.7 ±0.1	83.8 ±0.1	94.6 ±0.1	70.5 ±0.8	74.5 ±0.8	73.0 ±0.8	75.1 ±0.8
DoTR-S	33.3M	94.6 ±0.1	93.9 ±0.1	84.8 ±0.1	94.9 ±0.1	71.4 ±0.8	75.2 ±0.8	73.7 ±0.8	75.9 ±0.8
DoTR-B	107.3M	94.6 ±0.1	93.9 ±0.1	84.7 ±0.1	94.8 ±0.1	71.7 ±0.8	75.7 ±0.8	74.2 ±0.8	76.5 ±0.8
DoTR-L	343.9M	94.7 ±0.1	94.0 ±0.1	85.1 ±0.1	94.9 ±0.1	71.5 ±0.8	75.5 ±0.8	73.9 ±0.8	75.7 ±0.8
pretrained + fine-tuned									
DoTR-S-FT	33.3M	94.7 ±0.1	94.0 ±0.1	84.8 ±0.1	94.9 ±0.1	72.6 ±0.8	76.6 ±0.8	75.2 ±0.8	76.3 ±0.8
DoTR-B-FT	107.3M	**94.9** ±0.1	**94.2** ±0.1	**85.2** ±0.1	**95.1** ±0.1	73.0 ±0.8	**76.9** ±0.8	**75.5** ±0.8	**77.1** ±0.8
DoTR-L-FT	343.9M	**94.9** ±0.1	**94.2** ±0.1	85.1 ±0.1	95.0 ±0.1	**73.1** ±0.8	**76.9** ±0.8	**75.5** ±0.8	76.9 ±0.8
empirical upper bound									
GT + MiAE-B	–	100.0	100.0	100.0	100.0	83.0	83.0	83.0	86.0
parsing methods									
DoTR-B-CA	107.0M	95.0 ±0.1	94.3 ±0.1	85.6 ±0.1	95.3 ±0.1	–	–	–	–
Chainsaw	0.6M	92.5 ±0.1	89.9 ±0.2	78.6 ±0.2	91.4 ±0.1	–	–	–	–
UniDoc	–	88.9 ±0.1	86.1 ±0.2	68.2 ±0.2	90.2 ±0.1	–	–	–	–
Chainsaw + UniDoc	–	91.0 ±0.1	87.3 ±0.2	75.9 ±0.2	90.0 ±0.1	–	–	–	–
Chainsaw*	0.6M	90.5 ±0.1	87.4 ±0.2	79.5 ±0.1	91.5 ±0.1	–	–	–	–
Merizo*	51.8M	90.4 ±0.1	86.2 ±0.2	78.5 ±0.1	92.0 ±0.1	–	–	–	–

Table 8: **Benchmark results on TEDBench-Joint.** Void residues are scored as NDRs rather than ignored. $\pm$ is as in Table 7.

		Parsing metrics				Unified metrics			
Model	**Size**	**IoU**	**FracParsed**	**Boundary**	**NDO**	**mAP**	**AP$_{50}$**	**AP$_{75}$**	**PQ**
supervised from scratch									
DoTR-N	2.4M	90.5 ±0.1	85.4 ±0.2	76.3 ±0.2	83.2 ±0.2	61.4 ±0.8	66.9 ±0.8	64.4 ±0.8	67.2 ±0.8
DoTR-T	9.7M	91.4 ±0.1	86.5 ±0.2	78.4 ±0.2	84.0 ±0.2	67.0 ±0.8	72.3 ±0.8	69.9 ±0.8	70.1 ±0.8
DoTR-S	33.3M	91.5 ±0.1	86.7 ±0.2	79.2 ±0.2	84.2 ±0.2	68.2 ±0.8	73.3 ±0.8	71.1 ±0.8	71.0 ±0.8
DoTR-B	107.3M	91.6 ±0.1	86.7 ±0.2	79.4 ±0.2	84.2 ±0.2	68.4 ±0.8	73.8 ±0.8	71.4 ±0.8	71.6 ±0.8
DoTR-L	343.9M	91.5 ±0.1	86.6 ±0.2	79.6 ±0.2	84.1 ±0.2	67.9 ±0.8	73.3 ±0.8	70.8 ±0.8	71.0 ±0.8
pretrained + fine-tuned									
DoTR-S-FT	33.3M	91.6 ±0.1	86.9 ±0.2	79.3 ±0.2	84.3 ±0.2	69.2 ±0.8	74.6 ±0.8	72.3 ±0.9	71.4 ±0.8
DoTR-B-FT	107.3M	91.7 ±0.1	86.9 ±0.2	**79.8** ±0.2	84.4 ±0.2	**70.0** ±0.8	75.1 ±0.8	**73.1** ±0.8	**72.1** ±0.8
DoTR-L-FT	343.9M	**91.9** ±0.1	**87.3** ±0.2	79.7 ±0.2	**84.5** ±0.2	69.8 ±0.8	**75.2** ±0.8	72.7 ±0.8	71.9 ±0.8
empirical upper bound									
GT + MiAE-B	–	100.0	100.0	100.0	100.0	83.0	83.0	83.0	86.0
parsing methods									
DoTR-B-CA	107.0M	91.8 ±0.1	86.7 ±0.2	79.9 ±0.2	83.4 ±0.2	–	–	–	–
Chainsaw	0.6M	90.4 ±0.1	85.0 ±0.2	74.9 ±0.2	82.0 ±0.2	–	–	–	–
UniDoc	–	84.7 ±0.1	75.1 ±0.2	64.6 ±0.2	74.7 ±0.3	–	–	–	–
Chainsaw + UniDoc	–	88.7 ±0.1	81.9 ±0.2	72.5 ±0.2	80.5 ±0.2	–	–	–	–
Chainsaw*	0.6M	85.9 ±0.1	77.8 ±0.2	76.1 ±0.1	77.1 ±0.3	–	–	–	–
Merizo*	51.8M	87.9 ±0.1	81.1 ±0.2	75.4 ±0.1	76.4 ±0.2	–	–	–	–

Table 9: **End-to-end DoTR vs. two-stage methods.** All void residues are excluded from scoring. $\pm$ denotes $1.96\times$ the bootstrap standard error, from 1000 resamples of the $96{,}864$ test chains drawn with replacement. Point estimates are the values measured on the test set. DoTR-B-CA is the class-agnostic variant of DoTR-B-FT, trained with a single objectness class.

Model	Size	mAP	AP_{50}	AP_{75}	PQ
end-to-end					
DoTR-B-FT	107.3M	**73.0** ± 0.8	**76.9** ± 0.8	**75.5** ± 0.8	77.1 ± 0.8
two-stage					
DoTR-B-CA + MiAE-B	209.7M	72.5 ± 0.8	76.4 ± 0.8	74.9 ± 0.8	**77.9** ± 0.8
DoTR-B-FT + MiAE-B	209.9M	69.6 ± 0.8	73.3 ± 0.8	71.9 ± 0.8	**77.9** ± 0.8
Chainsaw + MiAE-B	103.3M	65.5 ± 0.8	72.7 ± 0.8	67.8 ± 0.9	72.9 ± 0.8
Merizo + MiAE-B	154.5M	57.8 ± 0.8	68.3 ± 0.9	60.1 ± 0.9	68.1 ± 0.8
UniDoc + MiAE-B	—	—	—	—	69.0 ± 0.7
Chainsaw + UniDoc + MiAE-B	103.3M	—	—	—	70.6 ± 0.8
empirical upper bound					
GT + MiAE-B	—	83.0	83.0	83.0	86.0

Table 10: **End-to-end DoTR vs. two-stage methods.** Void residues are scored as NDRs rather than ignored. $\pm$ is as in Table 9.

Model	Size	mAP	AP_{50}	AP_{75}	PQ
end-to-end					
DoTR-B-FT	107.3M	**70.0** ± 0.8	**75.1** ± 0.8	**73.1** ± 0.8	72.1 ± 0.8
two-stage					
DoTR-B-CA + MiAE-B	209.7M	69.3 ± 0.8	74.6 ± 0.8	72.3 ± 0.9	71.9 ± 0.8
DoTR-B-FT + MiAE-B	209.9M	66.7 ± 0.8	71.6 ± 0.8	69.6 ± 0.8	**73.0** ± 0.8
Chainsaw + MiAE-B	103.3M	61.2 ± 0.8	69.5 ± 0.8	63.5 ± 0.9	67.8 ± 0.8
Merizo + MiAE-B	154.5M	51.9 ± 0.8	63.9 ± 0.8	54.9 ± 0.9	60.9 ± 0.7
UniDoc + MiAE-B	—	—	—	—	62.0 ± 0.7
Chainsaw + UniDoc + MiAE-B	103.3M	—	—	—	65.9 ± 0.8
empirical upper bound					
GT + MiAE-B	—	83.0	83.0	83.0	86.0

and test always use full chains. We select the checkpoint with the best validation mAP and train each configuration with a single seed.

Class-agnostic DoTR. To isolate the contribution of the classification objective, we train a class-agnostic variant using the recipe from Xu et al. (2023): the vocabulary is collapsed to a single domain class, every ground-truth label is set to zero, and nothing else is changed. The classification head therefore predicts domain versus no-object and the multi-way cross-entropy becomes a binary objectness loss, while the matching costs, loss weights, the no-object weight of, and all remaining settings stay as in the base model. As in both works, we do not simply set the classification weight to zero, which would leave the class head untrained and its predictions arbitrary. Lacking a class-aware selection metric, this run selects its checkpoint by class-agnostic length-weighted IoU, and we report only class-agnostic metrics for it.

Hyperparameter selection. The recipe was selected on the validation split with DoTR-S at a fixed crop of 512 residues, and then transferred unchanged to all the models. We compared two learning schedules from SAM (Kirillov et al., 2023) and DETR (Carion et al., 2020). The DETR schedule was better on our validation set and is the one we use for all experiments. Within the adopted learning schedule, we briefly tune the peak learning rate, the layer-wise learning rate decay of the pretrained encoder, the weight decay, and the training length on the validation set. The remaining values were adopted from the corresponding reference methods rather than tuned: the decoder widths and the absence of auxiliary losses on intermediate decoder blocks follow SAM (Kirillov et al., 2023), the stochastic depth rates follow SAM 2 (Ravi et al., 2025), and the loss and matching weights follow Mask2Former (Cheng et al., 2022)

Table 11: Ablation studies for DoTR. The default setting (shaded) is shared by every subtable: normalized sine decoder PE, full loss, no sequence input, 20 queries, 3 decoder layers and a 512-residue crop. BW denotes the weight for the no-object class. Ablating BW means using the weight of 1.0 as for any other class.

#queries	mAP	PQ
5	66.7	73.8
10	68.4	74.2
20	68.9	73.5
40	69.3	74.1
80	67.7	72.5

(a) Decoder queries

decoder PE	mAP	PQ
Sine	68.0	73.1
Norm. sine	68.9	73.5
None	67.6	73.4
RoPE	66.2	71.9

(b) Positional encoding

#layers	mAP	PQ
1	67.6	72.9
2	67.9	73.4
3	68.9	73.5

(c) Decoder depth

loss	mAP	PQ
Full	68.9	73.5
− DICE	69.7	73.1
− BW	56.9	65.9

(d) Loss terms

enc. input	mAP	PQ
Structure	68.9	73.5
+ sequence	68.4	73.5

(e) Sequence information

crop size	mAP	PQ
512	68.9	73.5
256–1024	71.4	75.9

(f) Crop size

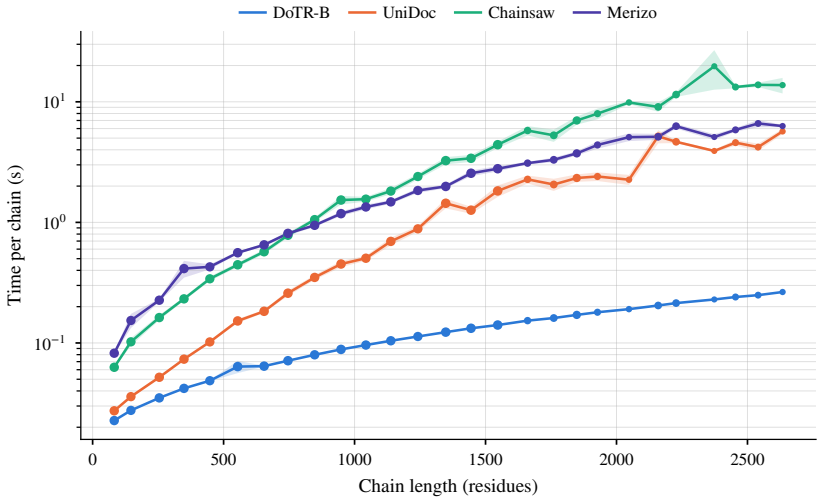

Figure 6: **Runtime comparison.** Time per chain from structure file to full domain annotation.

F.2 Parsing Baselines Experimental Details

Our model is compared against three parsing methods: Chainsaw (Wells et al., 2024), Merizo (Lau et al., 2023), and UniDoc (Zhu et al., 2023). All three baselines were originally developed on CATH domains in experimentally determined structures, where almost every residue belongs to a domain. Predicted structures break that assumption as they contain long disordered and non-domain regions (NDRs). Each method adapted differently. Merizo releases a checkpoint fine-tuned for AlphaFold models, together with a pipeline that re-segments iteratively and merges domains. Chainsaw keeps its CATH-trained model and adds post-processing heuristics to repair its output on predicted structures. UniDoc cannot handle NDRs at all, so they must be removed before or after the algorithm.

We sweep different post-processing options and thresholds for the baselines, and select the best models based on the validation set IoU. All selected models are run on the full TEDBench-Joint test split and scored with the same parsing evaluator as our own model. Each method is described in its original publication and we summarize here only our evaluation details.

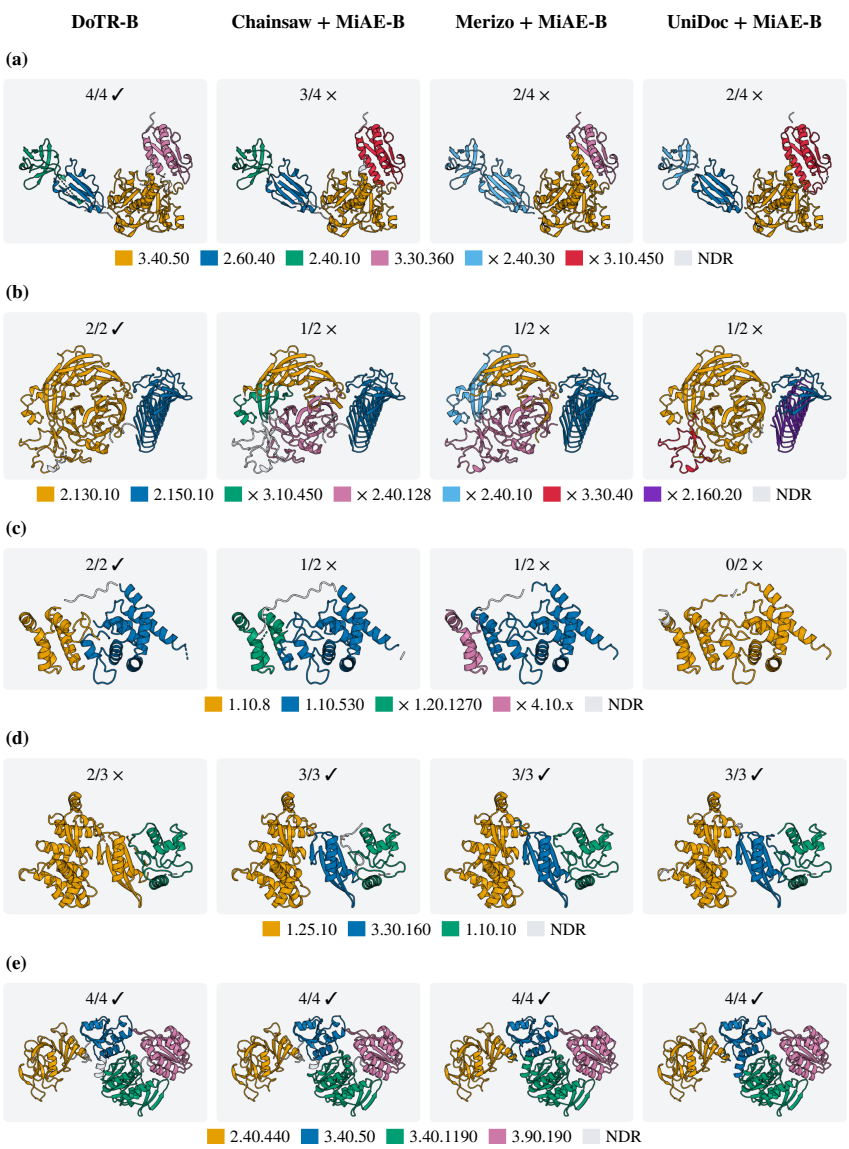

Figure 7: Additional examples on test chains: DoTR-B against the three two-stage pipelines (Chainsaw, Merizo and UniDoc, each followed by MiAE-B). (a)–(c) The two-stage pipelines make errors and DoTR-B is correct. (d) DoTR-B makes an error and all two-stage pipelines are correct. (e) All methods are correct.

Chainsaw, retrained. Chainsaw is trained on the TEDBench-Joint training split with the published protocol and void residues treated as NDR to match DoTR. Because it is not a released model, no convention fixes how its output should be post-processed. We compared three readouts on validation: no post-processing, Chainsaw's post-processing, and Chainsaw's post-processing followed by TED's filter. The setting without post-processing had the best validation performance, and is therefore the chosen setting.

Released checkpoints (Merizo*, Chainsaw*). TED already establishes the pipeline for running released checkpoints of Merizo and Chainsaw on AFDB structures. For Merizo, it enables iterative re-segmentation, a second pass without the NDRs found by the first, and TED's domain-merging step, followed by TED's fragment and minimum-domain filter. For Chainsaw this is its released checkpoint with its own post-processing enabled, followed by the same TED filter.

Table 12: **DoTR variants.** Each encoder prepends two geometric attention blocks to the listed Transformer layers. All variants use the same decoder: three two-way attention blocks of width 256 with 8 attention heads, a feed-forward dimension of 2048, and $Q = 20$ queries. The only exception is DoTR-N, which uses a smaller decoder with width of 128.

Model	# params	# layers	hidden dim	# heads	# v-heads	drop path
DoTR-N	2.4M	2	128	2	16	0.1
DoTR-T	9.7M	3	256	4	32	0.1
DoTR-S	33.3M	6	512	8	64	0.1
DoTR-B	107.3M	12	768	12	96	0.2
DoTR-L	343.9M	24	1024	16	128	0.3

Table 13: Training setting of DoTR. The same setting is used for every model size and for both training protocols, which differ only in the encoder initialization.

config	value
Optimization	
optimizer	AdamW
learning rate	2×10^{-4}
weight decay	0.1
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
effective batch size	64
learning rate schedule	step decay ($\times 0.1$ after $2/3$ of training)
warmup iterations	none
training epochs	60
stochastic depth	per model size (Table 12)
encoder initialization	random / MiAE-⟨size⟩
layer-wise lr decay	none
Data	
crop size	$\{256, 384, \ldots, 1024\}$, drawn per batch
data augmentation	Gaussian noise, $\mathcal{N}(0, 0.2^2)$
Loss	
classification loss weight λ_{cls}	2.0
mask loss weight λ_{bce}	5.0
Dice loss weight λ_{dice}	5.0
no-object weight $w_{\varnothing}$	0.1
matching cost weights (cls, bce, dice)	2.0, 5.0, 5.0
Evaluation	
mask threshold τ_{mask}	0.5
duplicate threshold τ_{ov}	0.8
model selection	best validation mAP
seed	42 (one seed per configuration)

UniDoc. UniDoc needs the most adaptation, since it does not handle NDRs. The two published evaluations of UniDoc on AlphaFold models each describe one pLDDT-based operation. Lau et al. (2023) remove residues below a pLDDT threshold before running UniDoc, while Wells et al. (2024) remove residues with pLDDT below 70 from its output. We therefore swept both directions over thresholds $\{50, 60, 70, 80\}$, with and without TED's filter, together with no NDR removal at all, and selected on the validation set. The selected configuration removes residues below pLDDT 60 from UniDoc's output and then applies TED's filter.

Chainsaw + UniDoc. Following TED's use of a learned parser to remove NDRs before UniDoc, we hand UniDoc the retrained Chainsaw's partition. We have chosen Chainsaw because it is the strongest out of all the baselines.

F.3 Two-stage Baselines Details

A two-stage pipeline parses the chain, cuts out each predicted domain, and classifies it independently into one of the $1{,}055$ fold classes. Only the labels and the detection scores come from the second stage, so every pipeline inherits the parsing metrics of its parser.

Classifier. All pipelines and the oracle share the MiAE-B + seq encoder from Chen et al. (2026a) (102.7M parameters) with a $1{,}055$-way head, fine-tuned for 50 epochs on the isolated ground-truth domains of the training split (AdamW, effective batch $1{,}024$, peak learning rate 1.6×10^{-3}, layer-wise learning-rate decay 0.8, weight decay 0.1). It reaches 92.4% top-1 accuracy and 86.0 macro-F1 on the ground-truth domains of the test split.

Detection score. Average precision ranks predictions by a score, which PQ and the parsing metrics do not use. We use the product of the parser confidence in the domain and the classifier's confidence in the label. Chainsaw outputs confidence over whole chains, so we take the chain confidence as the score for each domain in the chain. UniDoc produces no confidence, and is excluded from AP.

Oracle. The oracle replaces the parser with the ground-truth domains, following the empirical upper bound in object detection (Borji & Iranmanesh, 2019). Its parsing metrics are 100 by construction, its unified metrics are the ceiling the classifier imposes on any pipeline built from it, and every ground-truth domain is matched exactly, so all AP thresholds coincide. It is unchanged across the void settings, since no uncertain residue lies inside a ground-truth domain.

G Inference Runtime Setup

The timer spans one protein from its structure file to the method's final domain annotation, with models and databases loaded outside the loop. This is the boundary used by Chainsaw's released benchmark script (Wells et al., 2024). Process start, imports, checkpoint, database loading, and CUDA initialization are excluded for every method. The multi-stage pipelines pass intermediate results between stages through the file system, and we do not charge them for reading those results. DoTR-B performs no intermediate I/O, so these exclusions benefit only the baselines. Foldseek is a batch database search with no per-protein mode, and we measure it as it is deployed: one batch per length bin, from which a per-protein cost is obtained. Because this term is measured per bin rather than per protein, it contributes nothing to the within-bin standard error of the TED curve.

Sample and reporting. Proteins are grouped into 100-residue bins up to $2{,}661$ residues, with at most 30 sampled per bin. Each plotted point is the mean over the proteins of one bin and the shaded band is its standard error. Bins above $1{,}600$ residues contain fewer than 30 proteins because the benchmark holds no more, so single proteins move those points. Proteins are drawn from all splits rather than the test split alone, which does not contain enough long chains to populate the upper bins. Inference cost is a function of length, and these experiments do not calculate any accuracy metrics.

Execution. All measurements were made on one exclusive node with a single H100 GPU and eight pinned CPU threads, at batch size one. Each method is run with released code and the original software environment. Two-stage runtime is the sum of the parser and the classifier, and TED's is the sum of its pipeline stages.

H Limitations and Future Work

Our work has several limitations. Our model is currently restricted to closed-set classification, where it can only predict fold classes seen during training. This task could be reformulated as open-world domain segmentation, similar to computer vision. Under this formulation, the models could be used to discover novel domains by grouping similar fold classes together. Second, while DoTR predicts domain masks and fold labels, its outputs are not directly coupled to evolution or function. Prediction at deeper levels of the CATH hierarchy could open up further applications in structural

biology. Finally, although DoTR achieves strong performance, it is the first model of this kind, and further development is now enabled with the introduction of a large-scale benchmark.