Appendix

A Experimental setting

Data Descriptions for Different Stages of HTP. (1) For the single-protein sequence level, we employ the state-of-the-art ESM-2 [25], which outperforms all tested single-sequence protein language models across a wide range of structure prediction tasks and enables atomic resolution structure prediction. It is trained on 86 billion amino acids across 250 million protein sequences spanning evolutionary diversity. Specifically, ESM uses UniRef50, September 2021 version. The training dataset was partitioned by randomly selecting 0.5% ($\approx$ 250,000) sequences to form the validation set. The training set has sequences removed via the procedure described in Hie et al. [48]. ESM-2 runs MMseqs search to obtain the query and target databases. All train sequences which match a validation sequence with 50% sequence identity under this search are removed from the train set. The details of the ESM series can be found in `https://github.com/facebookresearch/esm`.

(2) For the antibody sequence level, we use the Observed Antibody Space database (OAS) [32] and its succeeding update [33] as the pretraining data. It currently contains over one billion sequences, from over 80 different studies that cover diverse immune states, organisms, and individuals, which can be downloaded from its official website at `https://opig.stats.ox.ac.uk/webapps/oas/`. We upload the processed paired data in `https://pan.baidu.com/s/18lB8gl9Maf0nnNPIw83ZzA?pwd=1212` (password: 1212) as well as the unpaired data in `https://pan.baidu.com/s/161gU8fso6rz6-QGfNoCoHQ?pwd=96uF` (password: 96uf).

(3) For the protein-protein complex structure level, we leverage the Database of Interacting Protein Structures (DIPS) [35]. It is a large protein complex structure dataset than existing antibody-antigen complex structure datasets and is mined from the Protein Data Bank [36]. We attain the database from Atom3d in Zendo `https://zenodo.org/record/4911102`, which is a collection of both novel and existing benchmark datasets spanning several key classes of biomolecules. Referring to Atom3d, we split protein complexes by sequence identity at 30%, resulting in train/validation/test sets with 87,303/31,050/15,268 instances.

(4) For the antibody-antigen complex structure level, we select all available antibody-antigen protein complexes from SAbDab [17] at `https://opig.stats.ox.ac.uk/webapps/newsabdab/sabdab/`, leading to a dataset containing 9,823 structures. CDRs are identified using the antibody numbering program AbRSA [49]. Following the setting in [13], the chosen data points are divided into training and test data based on their release date and CDR sequence identity. To be explicit, the test split contains protein structures released after December 24, 2021, as well as structures with any CDR similar to those released after this date with sequence identity higher than 50%. Antibodies in the test set are further clustered with 50% CDR sequence identity to remove duplicates, finally leading to 21 antibody-antigen structures. The training and validation splits just include complexes not involved during the curation of the test split. After that, we randomly split the remaining complexes with a ratio of 90% and 10% into the training and validation sets.

Implementation Details. HPT is implemented in PyTorch and PyTorch Geometric packages. For all four training stages, we leverage an Adam optimizer [50] with a weight decay of 1e-15. All experiments are run on multiple A100 GPUs, each with a memory storage of 80G.

(1) For ESM-2 in the single-protein sequence level training, we adopt a middle-size version, which has a parameter number of 150M, 30 layers, and a hidden dimension of 640. Besides, we append the ESM-2 with a three-layer perceptron to forecast the residue type for MLM.

(2) For the antibody sequence level training, we use a batch size of 2 to avoid the out-of-memory error and 4 workers to load the data. The number of epochs is 100 and starting learning rate is 1e-5. Apart from that, we utilize a ReduceLROnPlateau scheduler with a factor of 0.6, patience of 5 epochs, and a minimum learning rate of 1e-7.

(3) For the protein-protein complex structure level training, we use a batch size of 32, 1000 epochs, and 4 works to speed up data loading. The starting learning rate is 1e-4, and a ReduceLROnPlateau scheduler is utilized to adjust the learning rate automatically with a factor of 0.6 and patience of 3 epochs. We adopt a distance threshold of 8.0Åto determine the connection between different graph nodes (*i.e.*, the alpha carbon of each residue). As for the loss weight balance, we set $\lambda = 1$.

(4) For the antibody-antigen complex structure level training, we also adopt the distance threshold of 8.0Åto build the graph connection. For random initialization of CDR coordinates, we use a noise of $\epsilon = 0.1$. As for the other important hyperparameters, we use a grid search mechanism to find the optimal combination. Notably, the geometric neural networks used in the third and fourth levels are matched to each other. If we alter the setting of GGNNs in the antibody-antigen complex structure level training, we need to retrain it in the protein-protein complex structure level first. The entire hyperparameter search space is depicted in Table 4.

Table 4: Hyperparameters setup for HTP.

Hyperparameters Search Space	Symbol	Value
Training Setup		
Epochs	–	[100, 500, 1000]
Batch size	–	[32, 64, 128]
Learning rate	–	[1e-4, 5e-5, 1e-6, 1e-7]
Warmup	–	[Yes, No]
Warmup epochs	–	[10, 20]
Loss Balance weight for Coordinates and Residue Types	λ	[0.1, 0.3, 0.5, 0.7]
GNN Architecture		
Dropout rate	–	[0.1, 0.2]
Number of GNN layers	L	[2, 4, 6]
Tanh activation function	–	[Yes, No]
Coordinate Normalization	–	[Yes, No]
The hidden dimension of node representations	–	[320, 640]
The hidden dimension of edge representations	–	[16, 32, 64]

Reproduction of Baselines. Concerning the implementation of several baseline methods, we use the official repositories for conditional RefineGNN (`https://github.com/wengong-jin/RefineGNN/`), HERN (`https://github.com/wengong-jin/abdockgen`), DiffAb (`https://github.com/luost26/diffab`). To reproduce the performance of existing antibody-specific pretrained PLMs, we download the code from `https://github.com/alchemab/antiberta` for AntiBERTa and `https://github.com/oxpig/AbLang` for AbLang. In our comparison, we directly use their pretrained residue features as the input for GGNNs without any fine-tuning.

Code Availability. All relevant Python code to reproduce the results in our paper is stored in GitHub repository at `https://github.com/smiles724/HTP`.

B Additional Results

B.1 Ablation Study

We investigate the effectiveness and necessity of each component of our HTP. As shown in Table 5, the removal of protein-protein complex structure level induces performance detriment, where RMSD increases from 2.06 to 2.49. Moreover, we implement a variant of HTP by replacing features obtained by pretrained PLMs with learnable embedding features, whose performance is worse than HTP. To be concise, AAR declines from 40.98 to 25.31, and RMSD rises from 2.06 to 2.65. In summary, our HTP brings significant relative improvements of 78.56% in AAR, 41.97% in RMSD, and 2.94% in TM-Score. This phenomenon strongly supports the superiority of our approach over existing naive co-design algorithms that are trained only on antibody-specific structure data.

C Related Work

Antibody Design. The majority of old-school computational approaches for antibody design are based on sampling algorithms over hand-crafted and statistical energy functions to iteratively modify protein sequences and structures [7, 9, 10, 51, 52]. These physics-based algorithms are computationally expensive and prone to be stuck in local energy minimum, which triggers the

Table 5: Effects of each module, where SPS stands for the single-protein sequence level, PPCS denotes the protein-protein complex structure level, and AS represents the antibody sequence level. The last row computes the relative improvements of HTP over the primitive baseline without any protein data augmentation.

	SPS	AS	PPCS	SAbDab (CDR-H3) AAR (%) ↑	RMSD ↓	TM-Score
1	✗	✗	✗	22.95 ± 0.5	3.55 ± 0.01	0.9146 ± 0.003
2	✓	✗	✗	33.87 ± 0.8	2.77 ± 0.04	0.9450 ± 0.006
3	✓	✓	✗	$\underline{38.42 \pm 1.6}$	$\underline{2.49 \pm 0.03}$	$\underline{0.9538 \pm 0.004}$
4	✗	✗	✓	25.31 ± 0.7	2.95 ± 0.02	0.9391 ± 0.005
5	✓	✓	✓	$\mathbf{40.98 \pm 1.5}$	$\mathbf{2.06 \pm 0.03}$	$\mathbf{0.9621 \pm 0.005}$
Imp.	–	–	–	78.56%	41.97%	2.94%

adaptation of deep learning in this sub-field. The initial researchers [4, 5, 53, 54] use pure PLMs to generate protein sequences but disregard the available antigen structures.

To circumvent this, Jin et al. [11] introduce RefineGNN, the first co-design architecture that aims to neutralize SARS-CoV-2. Later, HERN [12] is proposed as a more general version for paratope docking and design, which opens the door to produce antibodies given arbitrary antigen structures. Subsequent efforts are spent in either modifying the generative style or utilizing more advanced deep learning architectures such as diffusion denoise probabilistic models (DDPMs). For example, DiffAb [13] achieves atomic-resolution antibody design with SO(3)-equivariance, while MEAN [15] corrects the autoregressive manner with a full-shot one to prevent low efficiency and accumulated errors during inference.

Protein Sequence Modeling. Sequence-based protein representation learning is mainly inspired by the field of natural language processing. A large body of early works concentrates on modeling individual protein families [55], solving problems like functional nanobody design [5]. Its success, then, motivates the prospective trend to model large-scale databases of protein sequences by means of unsupervised learning. This line of study targets capturing the biochemical and co-evolutionary knowledge that underlies a large-scale protein sequence corpus by self-supervised pertaining. Thanks to them, a number of pertaining objects have been explored such as the next amino acid prediction [4, 26], masked language modeling (MLM) [55, 23], pairwise MLM [56], contrastive predictive coding [57], conditional generation [58], and position-specific scoring matrix prediction [59]. In addition, another line [60, 61] is based on multiple sequence alignment (MSA), leveraging sequences within a protein family to seize the conserved and variable regions of homologous sequences. Notably, some schemes for protein sequence modeling also seek to incorporate structural information in either the pretraining stage [62, 63] or the finetuning stage [64].

The improvements in model scale and architecture are also crucial to the recent achievement of PLMs. Explicitly, Rao et al. [55] evaluate various PLMs in a panel of benchmarks and discover that multi-head attention outpaces the Potts model in contact prediction, even if using a single sequence for inference. Concurrently, Vig et al. [65] observe that specific attention heads of pretrained Transformers have straight correlations with protein contact. Others [26] investigate a variety of Transformer variants [66] and demonstrate that large Transformers can procure state-of-the-art features across diverse tasks. Apart from that, the latest ESM-2 [25] trains the largest PLM with 15B parameters and shows that as models are scaled, they learn information enabling the protein structure prediction at the resolution of individual atoms.

Protein Structure Learning. With the rapid advance of geometric deep learning, it has been increasingly attractive and challenging to represent and reason about structures of macromolecules in the 3D space. For the sake of encoding spatial information in protein structures including bond lengths and dihedral angles, numerous 3D geometric neural networks such as 3DCNN [67–70] or GNNs [45, 46, 71, 72] have been invented. They excel at capturing complex interactions between sets of amino acids [73] and attain pivotal Euclidean geometry, *e.g.*, E(3) or SE(3)-equivariance and symmetry.

However, compared to protein sequences in databases like UniProt [74] or Pfam [75], the known structures in the PDB are scarce and hard to obtain. Therefore, it becomes an urgent need to develop structure-based mechanisms to efficiently learn protein representations with much less pretraining data. For instance, Hermosilla and Ropinski [18] use contrastive learning in terms of molecular substructures to help models understand protein structure similarity and functionality. Moreover, Chen et al. [76] propose a self-supervised framework that predicts angles and inter-residue distances. Additionally, Guo et al. [77] present a coordinate denoising score matching method. Wu et al. [19] put forward a novel prompt-based denoising conformation generative pretraining method based on the trajectories of molecular dynamics simulations. A recent attempt [34] makes a combination of both contrastive learning and self-prediction with more intriguing augmentation functions. Despite this progress, all of them are dealing with single-protein structures. No preceding studies have considered structure-based pretraining in the circumstance of multiple proteins. That is, how to pretrain on protein-protein complex, or more specifically, the antibody-antigen complex, remains unexplored.

D Limitations and Future Work

In spite of the promising progress of our HTP, there is still some space left for future explorations. First, more abundant databases can be exploited in our framework. For instance, AntiBodies Chemically Defined (ABCD) [78] is a large-sized antibody sequence database that can be used to enhance the capacity of protein language models at the second level. We do not use it in our work because our request for this database has not been approved by the authors so far. Secondly, we fix the language models during the last two levels of training (*i.e.*, levels that need complex structure prediction) for simplicity and use them as the node feature initializer. It might be beneficial if both PLM and the geometric encoder are tuned.