Miscellaneous Research on BERT

Interpretation, GAN, Adaptation

Boyuan Wang

2020-07-10

Overview

- BERTology Meets Biology: Interpreting Attention in Protein Language Models (arXiv)
  - Visualization of attention weights of BERT and layer-wise probing tasks on BERT layer output. Interpreting how attention capture the underlying structural properties of protein thru important tasks like Secondary Structure, Contact Map, Binding Sites.
- GAN-BERT: Generative Adversarial Learning for Robust Text Classification with a Bunch of Labeled Examples (ACL 2020)
  - Semi-supervised Gan for BERT fine-tuning. This paper shows that the needs for abundant labelled data can be drastically reduced with the help of Semi-supervised GAN and unlabelled task data.
- Don't Stop Pretraining: Adapt Language Models to Domains and Tasks (ACL 2020)
  - This paper investigates and shows that tailoring a pertained model to domain of target tasks helps performance. Moreover, task unlabelled data and augmentation can achieve comparative results with much less data and memory requirement than that of domain data.

Overview

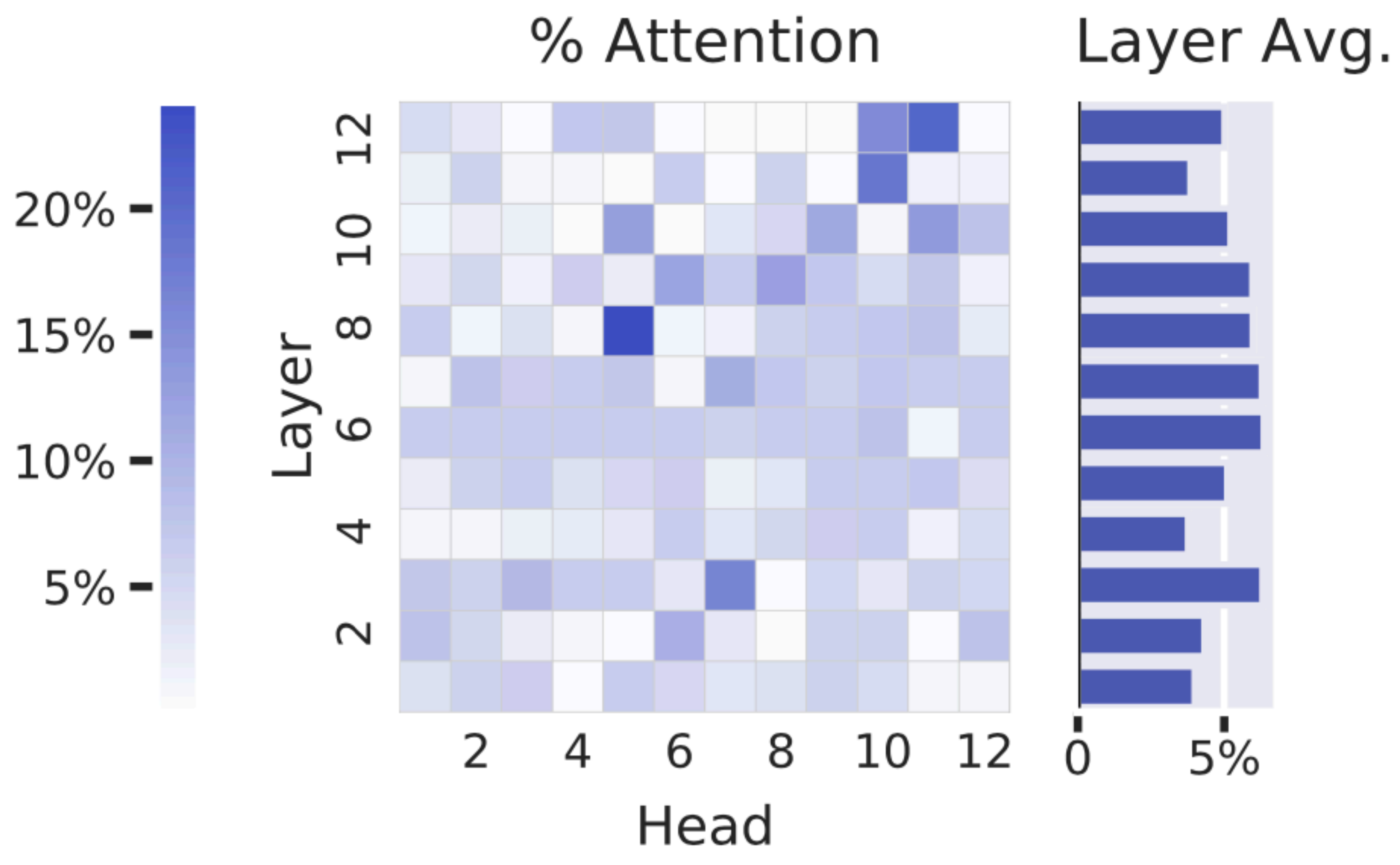
- BERTology Meets Biology: Interpreting Attention in Protein Language Models (arXiv)
  - Visualization of attention weights of BERT and layer-wise probing tasks on BERT layer output. Interpreting how attention capture the underlying structural properties of protein thru important tasks like Secondary Structure, Contact Map, Binding Sites.
- GAN-BERT: Generative Adversarial Learning for Robust Text Classification with a Bunch of Labeled Examples (ACL 2020)
  - Semi-supervised Gan for BERT fine-tuning. This paper shows that the needs for abundant labelled data can be drastically reduced with the help of Semi-supervised GAN and unlabelled task data.
- Don't Stop Pretraining: Adapt Language Models to Domains and Tasks (ACL 2020)
  - This paper investigates and shows that tailoring a pertained model to domain of target tasks helps performance. Moreover, task unlabelled data and augmentation can achieve comparative results with much less data and memory requirement than that of domain data.

Paper I - Motivation

- The study of protein is important for understanding of human health and the development of treatment for diseases.
- Self-supervised learning like BERT has gained big success for essential protein structure prediction tasks like Secondary Structure (SS), Contact Map and Binding Site.
- This paper adapts the interpretability research of Transformer Attention from NLP to analyze the inner work of BERT thru visualization and probing tasks.

Paper 1 - Method

- Attention map visualization



(f) Val

- Calculating scores for attention map, where f(i, j) is an indicator function for some task related property.

$$p_\alpha(f) = \sum_{x\in X}\sum_{i=1}^{|x|}\sum_{j=1}^{|x|} f(i,j)\alpha_{i,j}(x) \Big/ \sum_{x\in X}\sum_{i=1}^{|x|}\sum_{j=1}^{|x|}\alpha_{i,j}(x)$$

Paper 1 - Method

- Probing tasks:
  - Classification of layer-wise BERT output to analyze the flow of information in BERT.
  - Token-wise (SS, Binding Sites) probing: directly feed the output to classifier.
  - Token-pair probing (Contact Map): concat(elementary-wise diff, product of two tokens output).
  - Evaluation:

    F1 score for SS;

    Precision@L/20 for binding sites;

    Precision@L/5 for contact map;

Paper I - Dataset

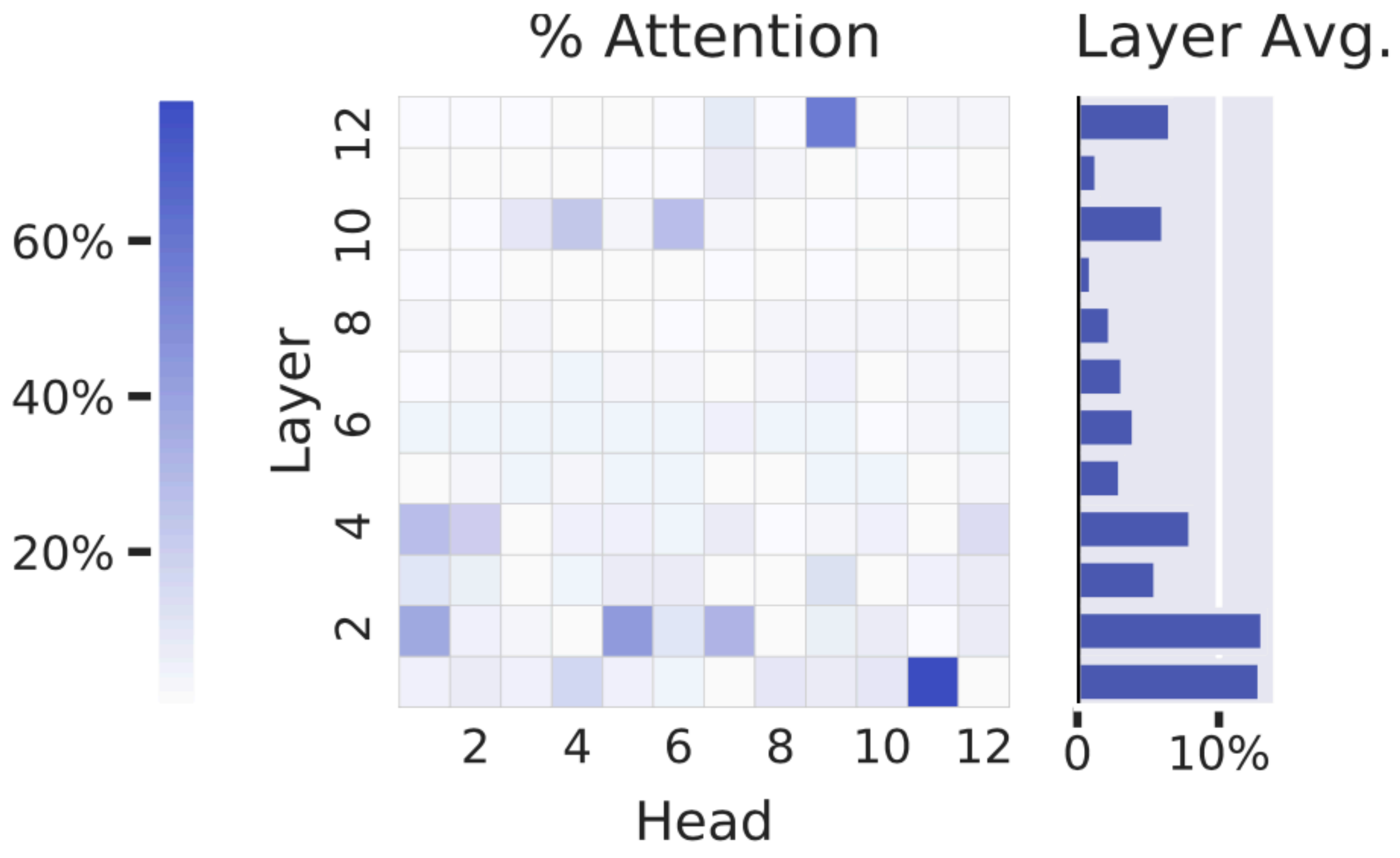
- Contact Map dataset: ProteinNet dataset, train 25299; valid 224
- Secondary Structure dataset: train 8678; valid 2170
- Binding Sites: created from SS dataset with annotation from Protein Data Bank's Web API. train 5734; valid 1418
- Attention Analysis: randomly 5000 draw from train sets.
- Probing task: full training sets.

Paper I - Experiments

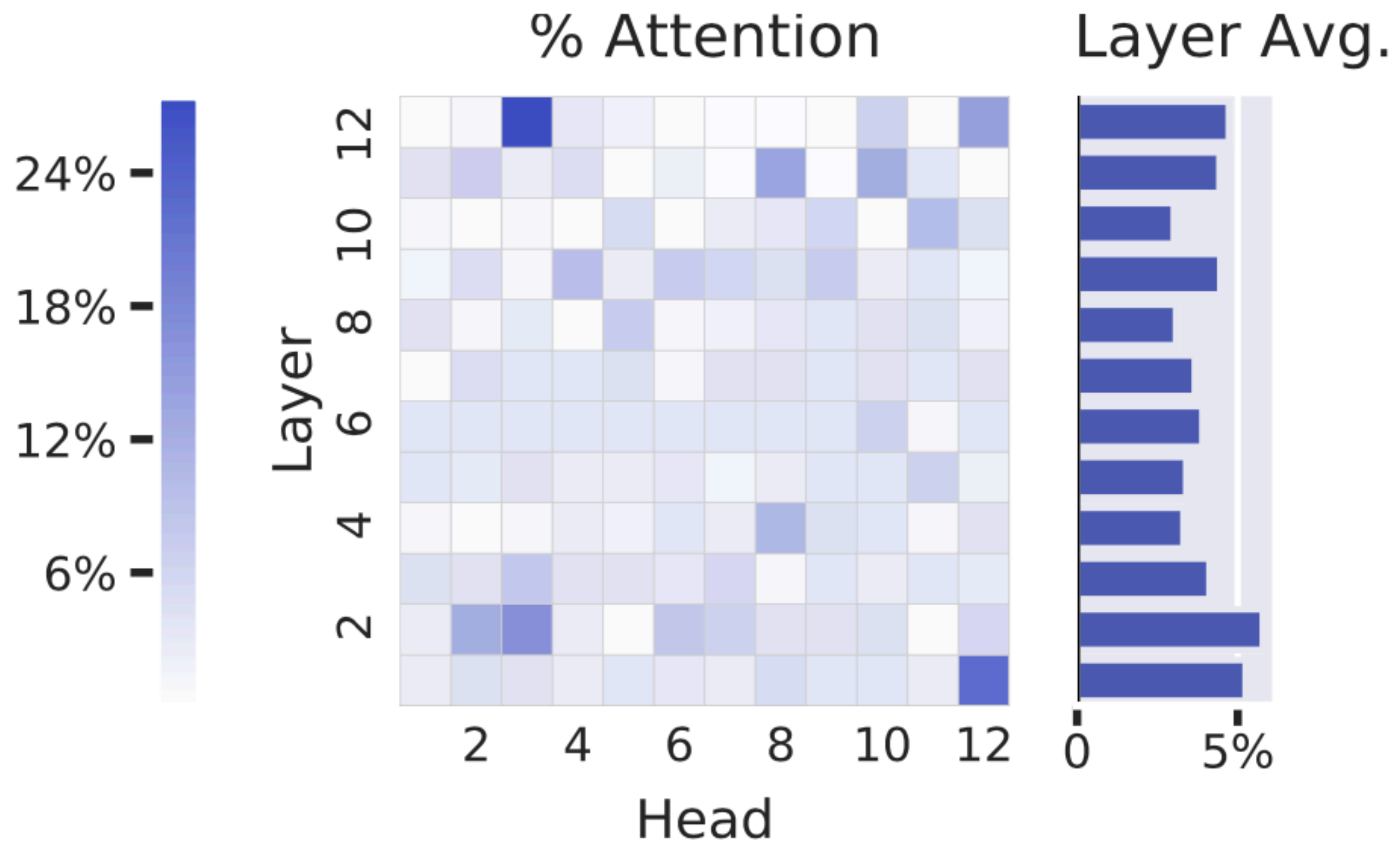
- Implementation:
  - Tape model as pretrained model.
  - Probing task: Linear FFN with softmax.
  - Attention weight between 0 and 1. Weight < 0.1 is filtered.
  - All protein sequences are truncated to length of 500, otherwise shorter.
  - GPU: Single Tesla V100 with 16GB memory.

Paper I - Analysis

- Amino Acids:
  - Attention heads specialize in certain types of amino acids.
  - Each block is the proportion of weight for one amino acids for one attention head.
  - The deeper the color of a block, the heavier the weight.



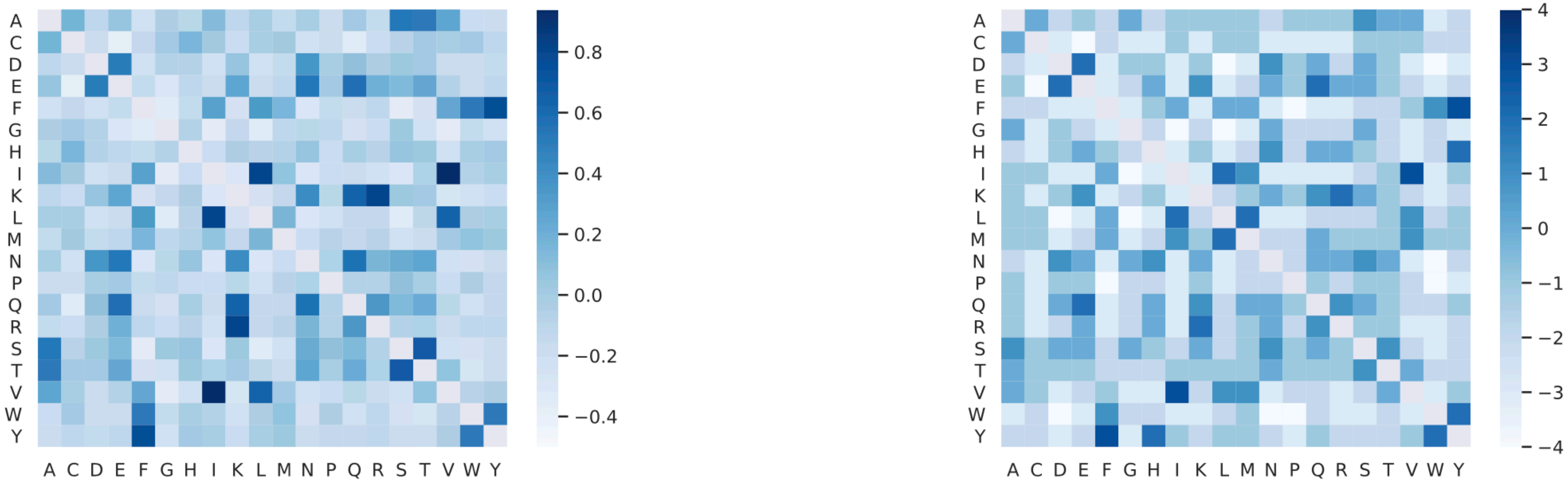
(a) Attention to amino acid *Pro*



(b) Attention to amino acid *Phe*

Paper I - Analysis

- Amino Acids:
  - Attention is consistent with substitution relationships.
  - Method: Pearson correlation between any two amino acid attention heat map to construct (a), and Pearson again between (a) and (b), (b) is the background substitution matrix (~profile).



(a) Attention similarity

(b) BLOSUM62 substitution scores

Figure 3: Comparison of attention similarity matrix with the substitution matrix. Each matrix entry represents an amino-acid pair (codes in Appendix B.1). The two matrices have a Pearson correlation of 0.80 with one another, suggesting that attention is largely consistent with substitution relationships.

Paper I - Analysis

- Contact Map:
  - Attention aligns strongly with contact maps in one attention head. (Surprising!)
  - Attention is a well-calibrated predictor of contact maps. Estimate contact prob by binning token pairs (i,j) to 10 bins based on attention weights and calculation the proportion that are actually in contact.

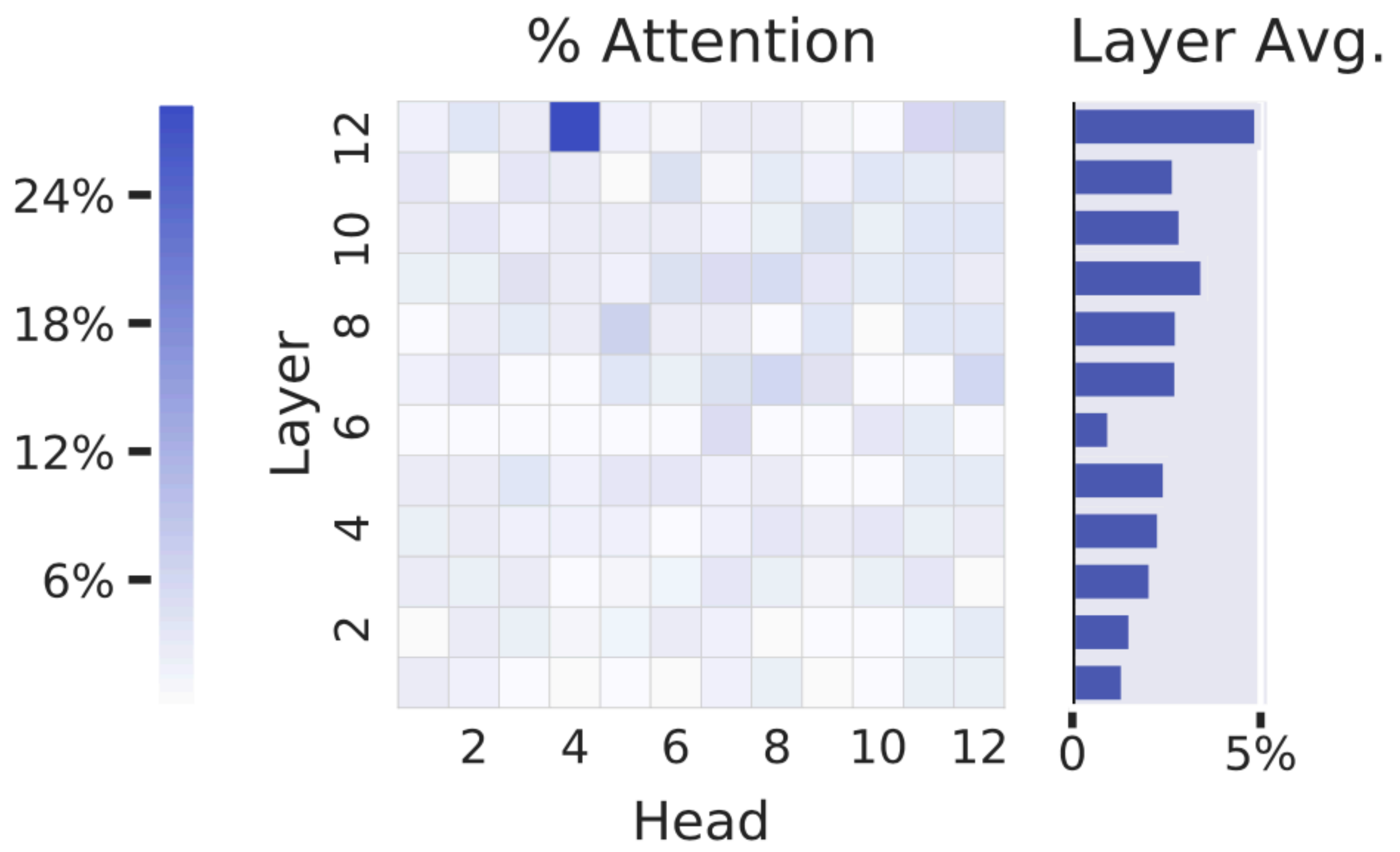


Figure 4: Percentage of each head's attention that is aligned with contact maps, averaged over a dataset, suggesting that Head 12-4 is uniquely specialized for contact prediction.

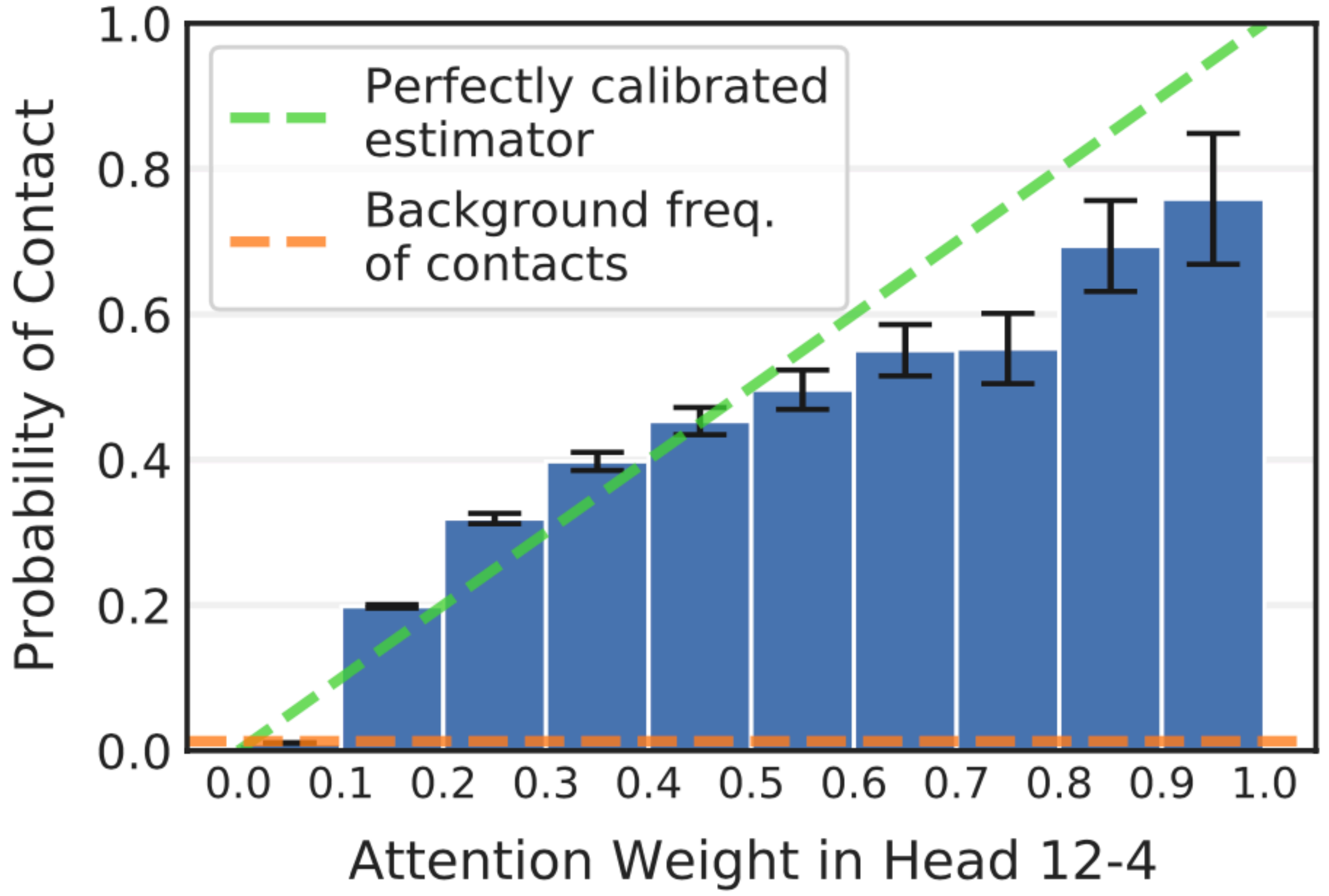


Figure 5: Probability two amino acids are in contact [95% confidence intervals], as a function of attention between the amino acids in Head 12-4, showing attention approximates a perfectly-calibrated estimator (green line).

Paper I - Analysis

- Binding Sites:
  - Attention targets binding sites, especially in the deeper layers.
  - Tokens often target binding sites from far away in the sequence. In Head 7-1, for example, the average distance spanned by attention to binding sites is 124 tokens.

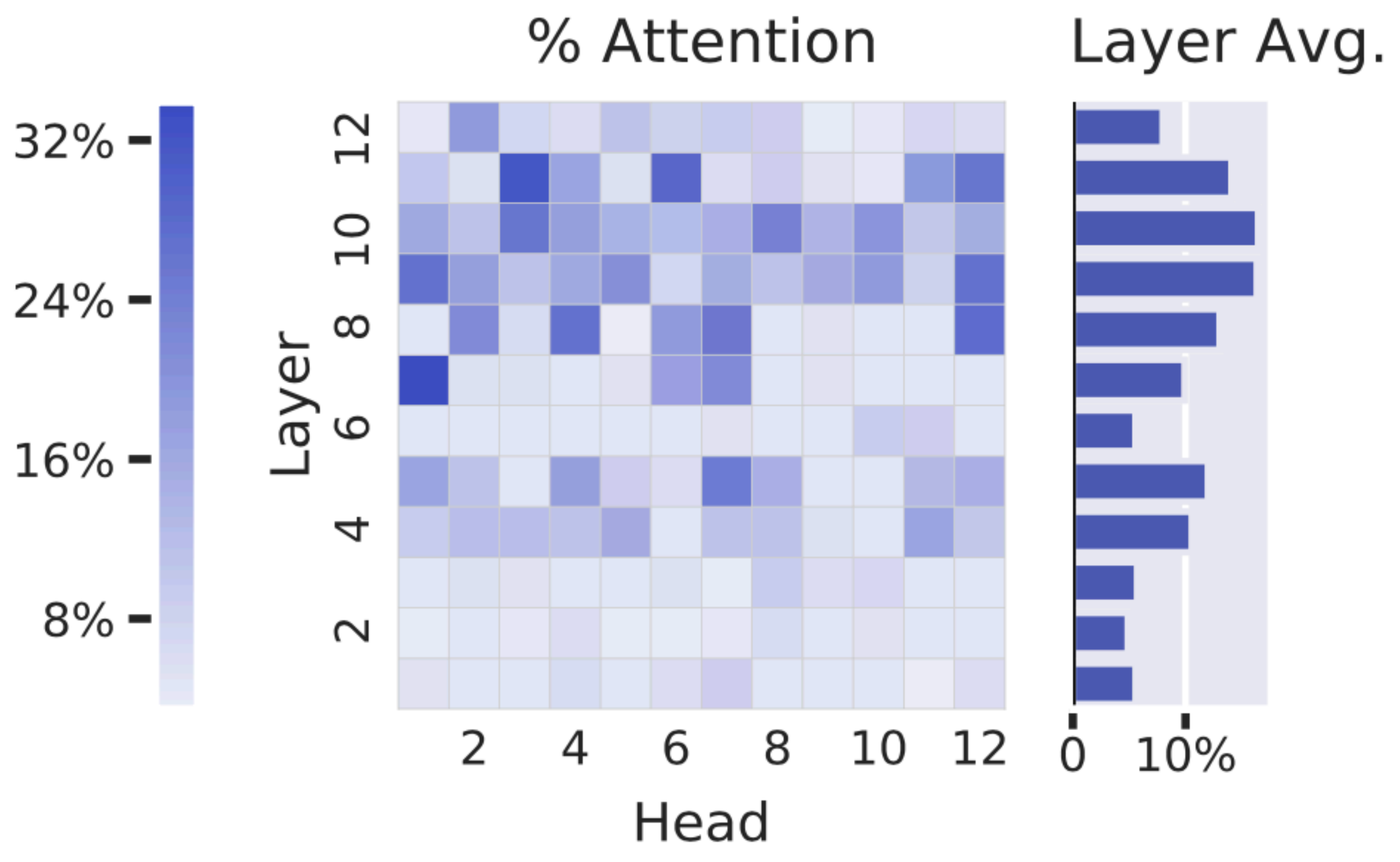
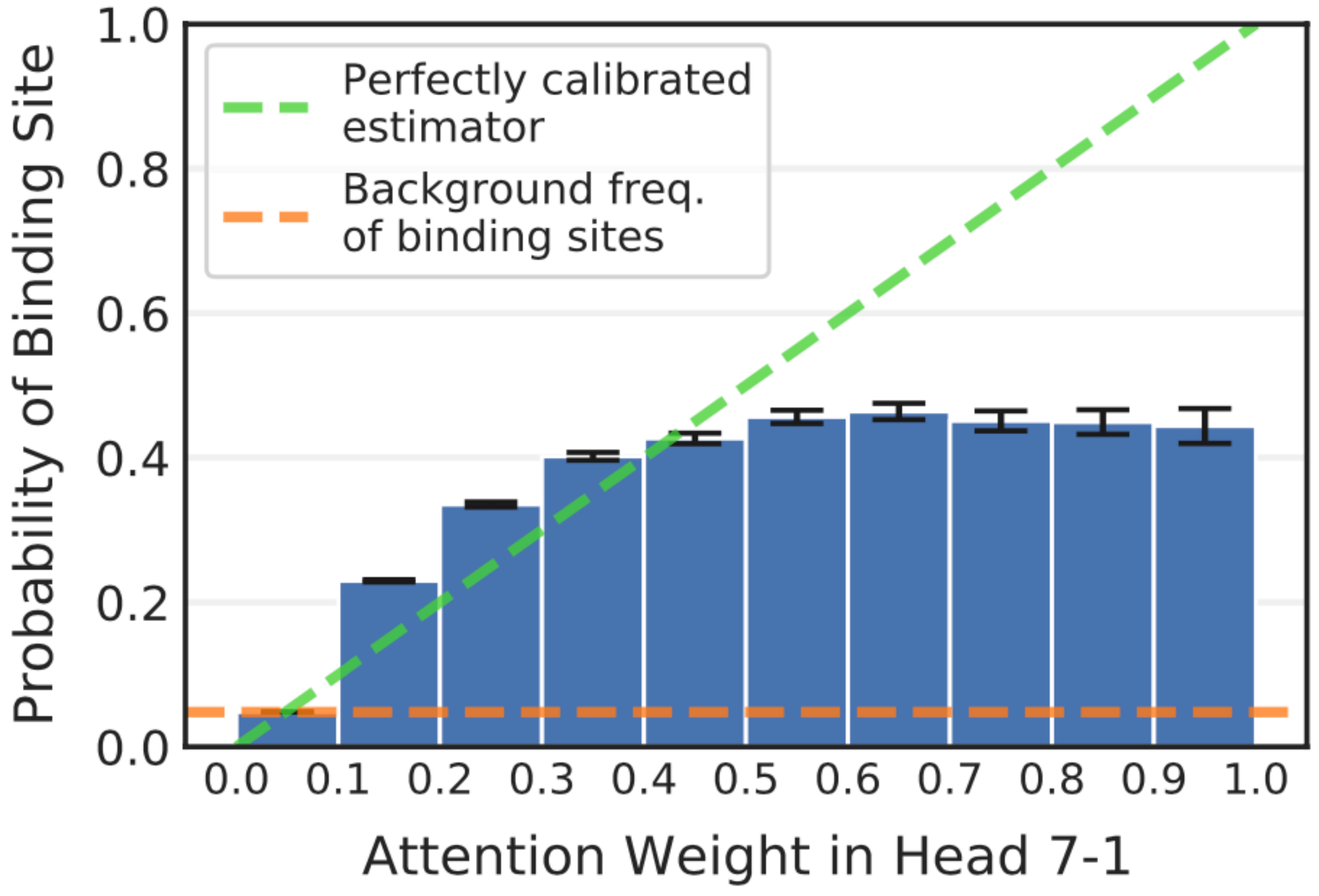
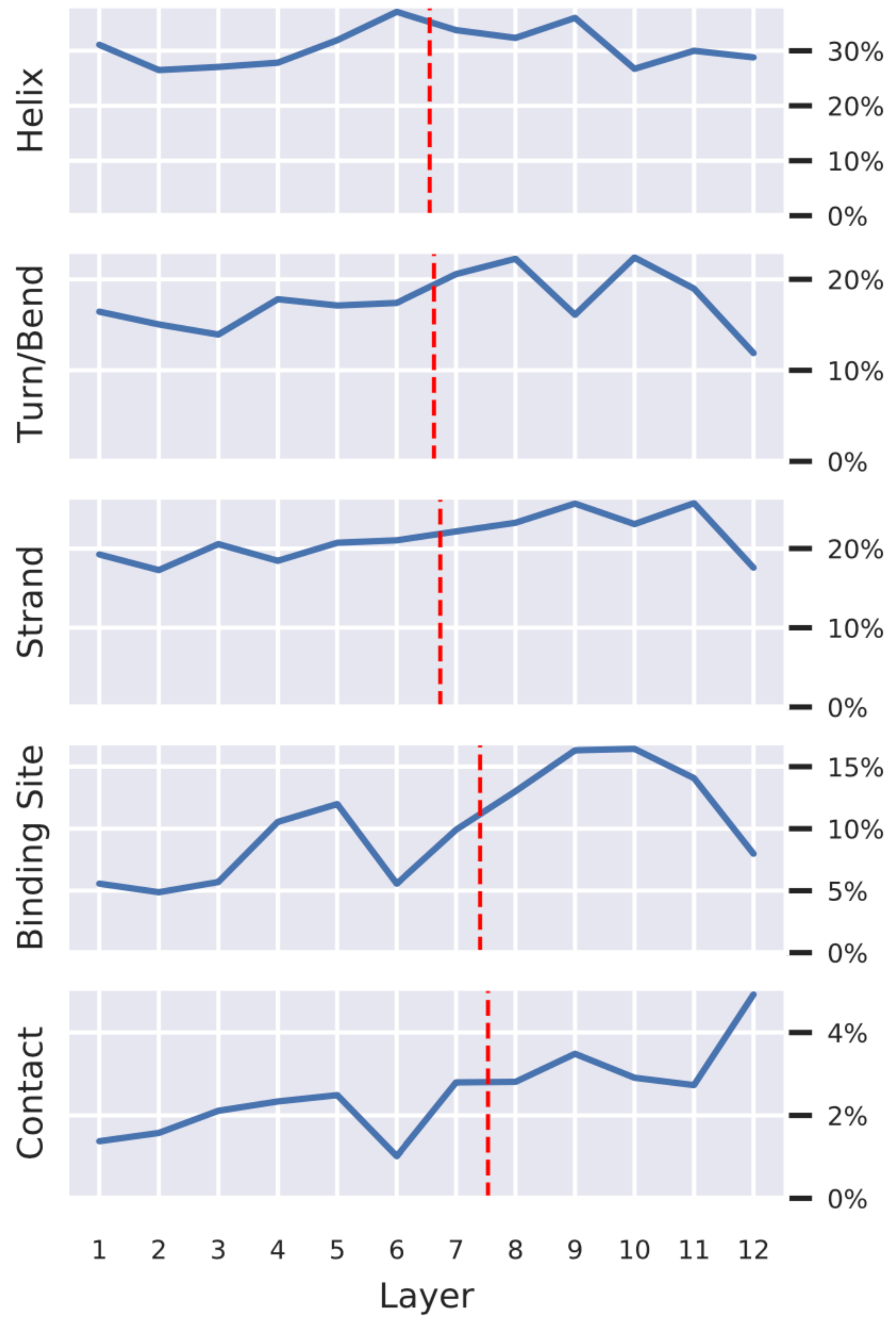


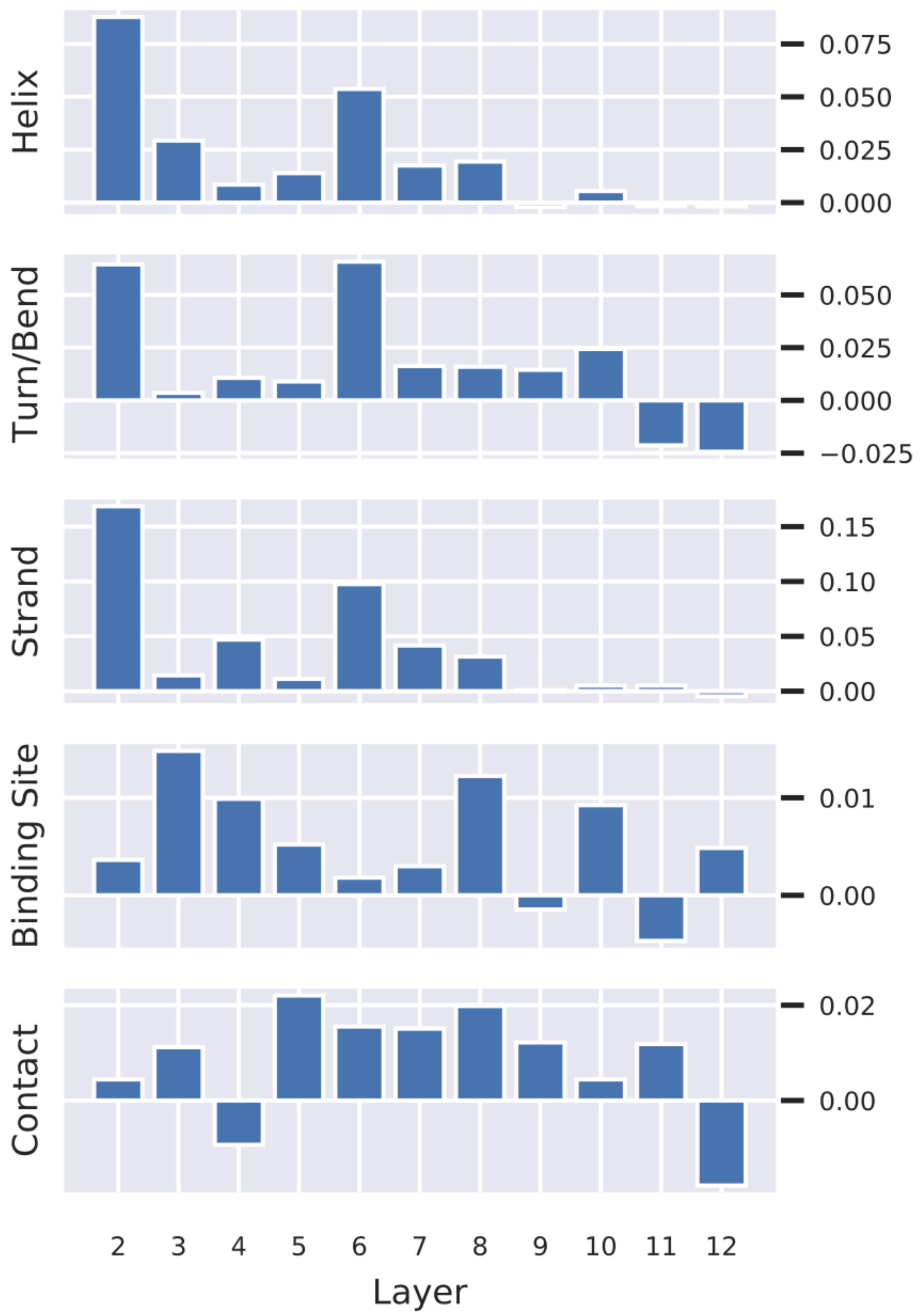
Figure 6: Percentage of each head's attention



Paper I - Analysis

- Layer-wise Probing analysis:
  - Attention targets higher-level properties in deeper layers.
  - The left figure is the percentage of attention focus across layers. The right figure is the layer wise probing result difference. High-level structural inform more attended at upper layers. (observe that last layer doesn’t perform the best)





Paper I - Conclusion

- This paper adapts NLP interpretation methods to protein sequence modeling.
- It shows how a Transformer language model recovers structural and functional properties of proteins and integrates this knowledge directly into its attention mechanism.
- It shows some insightful results about BERT learning local and global information of protein.

Overview

- BERTology Meets Biology: Interpreting Attention in Protein Language Models (arXiv)
  - Visualization of attention weights of BERT and layer-wise probing tasks on BERT layer output. Interpreting how attention capture the underlying structural properties of protein thru important tasks like Secondary Structure, Contact Map, Binding Sites.
- GAN-BERT: Generative Adversarial Learning for Robust Text Classification with a Bunch of Labeled Examples (ACL 2020)
  - Semi-supervised Gan for BERT fine-tuning. This paper shows that the needs for abundant labelled data can be drastically reduced with the help of Semi-supervised GAN and unlabelled task data.
- Don't Stop Pretraining: Adapt Language Models to Domains and Tasks (ACL 2020)
  - This paper investigates and shows that tailoring a pertained model to domain of target tasks helps performance. Moreover, task unlabelled data and augmentation can achieve comparative results with much less data and memory requirement than that of domain data.

Paper II - Motivation

- BERT achieves impressive results in many NLP tasks.
- However, many of these tasks are made of (sometimes hundreds of) thousands of examples.
- In reality, obtaining high quality annotated examples are expensive and time consuming. In contract, task specific unlabelled data are easier to collect.
- Semi-supervised GAN can utilize the unlabelled task data in generative adversarial setting to reduce the needs for labelled data (50 - 100 examples only) while still performs good.

Paper II - Method

- **Semi-supervised GAN**: Discriminator is trained over k+1 classes. True classes are in one of the (1,…,k) classes and generated data in class k+1. P_d is the data distribution.

- Discriminator loss: $L_{\mathcal{D}} = L_{\mathcal{D}_{\text{sup.}}} + L_{\mathcal{D}_{\text{unsup.}}}$

$$L_{\mathcal{D}_{\text{sup.}}} = -\mathbb{E}_{x,y\sim p_d} \log\left[p_{\text{m}}(\hat{y} = y|x, y \in (1, ..., k))\right]$$

$$L_{\mathcal{D}_{\text{unsup.}}} = -\mathbb{E}_{x\sim p_d} \log\left[1 - p_{\text{m}}\left(\hat{y} = y|x, y = k+1\right)\right]$$ (correct unlabelled true)

$$- \ \mathbb{E}_{x\sim\mathcal{G}} \log\left[p_{\text{m}}(\hat{y} = y|x, y = k+1)\right]$$ (correct generative fake)

- Generative loss: $L_{\mathcal{G}} = L_{\mathcal{G}_{\text{feature matching}}} + L_{\mathcal{G}_{unsup.}}.$

$$L_{G_{\text{feature matching}}} = \left\|\mathbb{E}_{x \ \sim \ p_d} f(x) \ - \ \mathbb{E}_{x \ \sim \ \mathcal{G}} f(x)\right\|_2^2$$

$$L_{\mathcal{G}_{unsup.}} = -\mathbb{E}_{x\sim\mathcal{G}} \log\left[1 - p_m\left(\hat{y} = y|x, y = k+1\right)\right]$$ (Minimize model probability P_m)

Paper II - Method

- **Gan-BERT**
- Implementation: G: 1-layer MLP with leaky Relu; D: 1-layer MLP with leaky Relu. Standard Gaussian noise. |U| = 100*|L|, data replicate log(|U|/|L|) for labelled data. Results averaged on 5 different shuffle of training data.

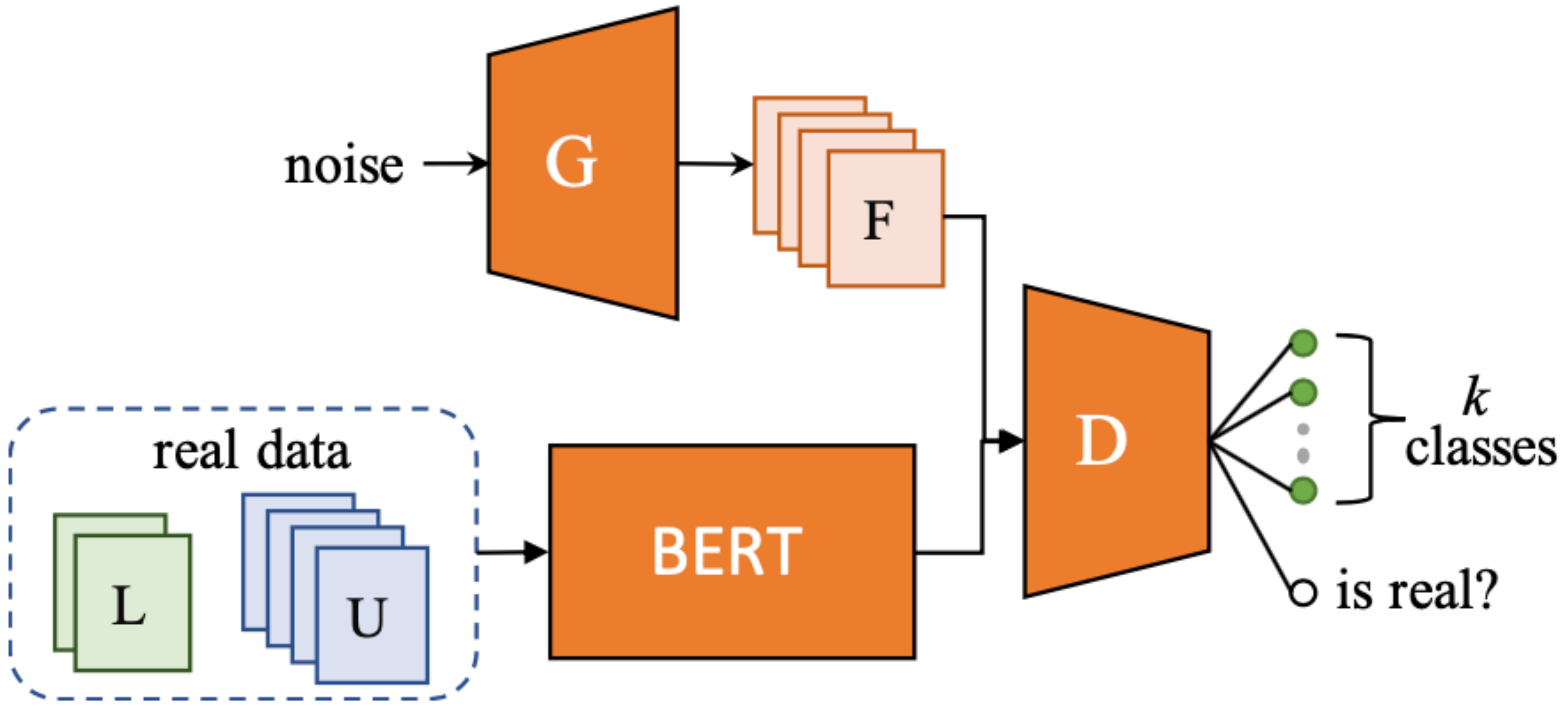
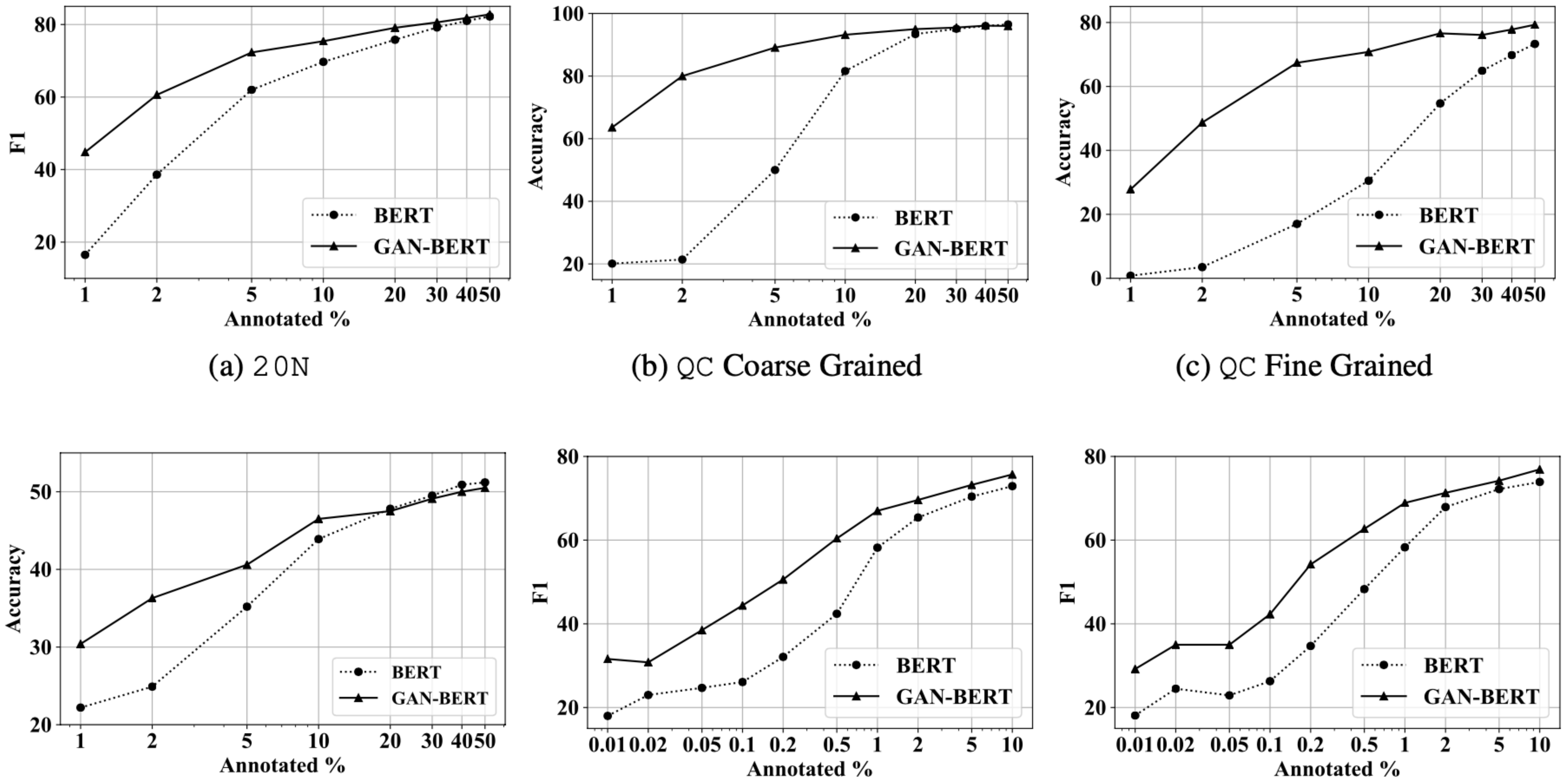


Figure 1: GAN−BERT architecture: $\mathcal{G}$ generates a set of fake examples F given a random distribution. These, along with unlabeled U and labeled L vector representations computed by BERT are used as input for the discriminator $\mathcal{D}$.

Paper II - Experiment

- 6 classification tasks results: 1% of annotated data means a few hundred examples.



(a) 20N

(b) QC Coarse Grained

(c) QC Fine Grained

(d) SST-5

(e) MNLI Matched

(f) MNLI Mismatched

Figure 2: Learning curves for the six tasks. We run all the models for 3 epochs except for 20N (15 epochs). The sequence length we used is: 64 for QC coarse, QC fine, and SST-5; 128 for both MNLI settings; 256 for 20N. Learning rate was set for all to 2e-5, except for 20N (5e-6).

Paper II - Conclusion

- Fine-tuning BERT with few labeled examples lead to unstable models.
- Semi-supervised GAN is advised to mitigate the problem.
- Generator abandoned for inference, so not introduced additional inference time.
- Sequence Labeling and QA haven’t been tested.

Overview

- BERTology Meets Biology: Interpreting Attention in Protein Language Models (arXiv)
  - Visualization of attention weights of BERT and layer-wise probing tasks on BERT layer output. Interpreting how attention capture the underlying structural properties of protein thru important tasks like Secondary Structure, Contact Map, Binding Sites.
- GAN-BERT: Generative Adversarial Learning for Robust Text Classification with a Bunch of Labeled Examples (ACL 2020)
  - Semi-supervised Gan for BERT fine-tuning. This paper shows that the needs for abundant labelled data can be drastically reduced with the help of Semi-supervised GAN and unlabelled task data.
- Don't Stop Pretraining: Adapt Language Models to Domains and Tasks (ACL 2020)
  - This paper investigates and shows that tailoring a pertained model to domain of target tasks helps performance. Moreover, task unlabelled data and augmentation can achieve comparative results with much less data and memory requirement than that of domain data.

Paper III - Motivation

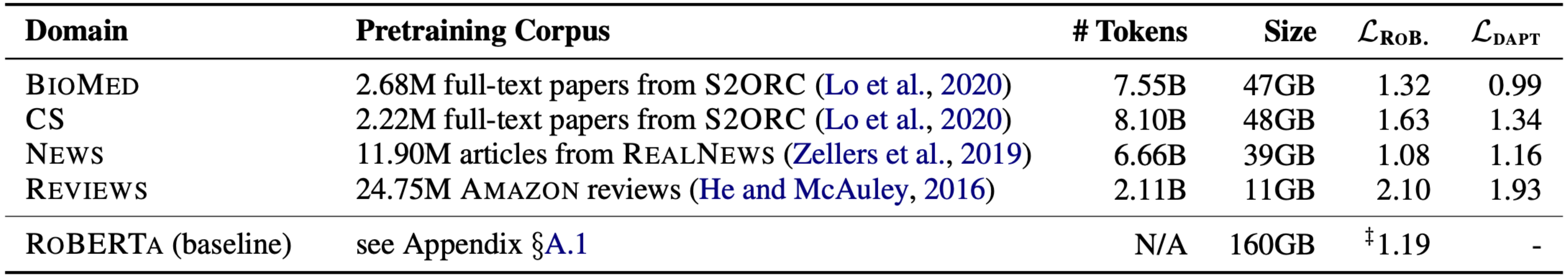
- RoBERTa was trained on over 160GB of uncompressed text, with sources ranging from English language encyclopedic and news articles, to literary works and web content.
- Representations learned by such models achieve strong performance across many tasks with datasets of varying sizes drawn from a variety of sources
- So, is it still helpful to tailor the model to the domain of the target task? Or the latest large pertained models works universally?

Paper III - Method

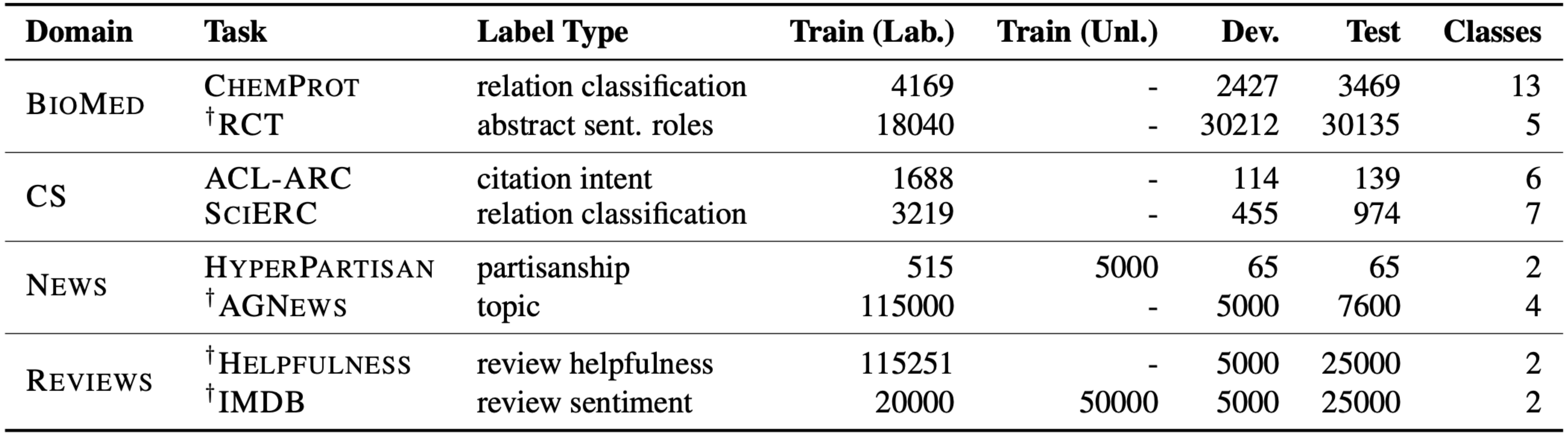
- Domain Adaptive Pretraining (DAPT): continuing pretraining on a large corpus of unlabelled domain specific text.
- Task Adaptive Pretraining (TAPT): continuing pretraining on smaller but directly task relevant unlabelled text from training data of a supervised task.
- Augmenting training data for TAPT: augment the supervised training set with unlabelled data from the training set distribution to obtain larger task specific text pool.

Paper III - Datasets

- The paper conducts extensive experiments on 4 domains and 8 tasks.



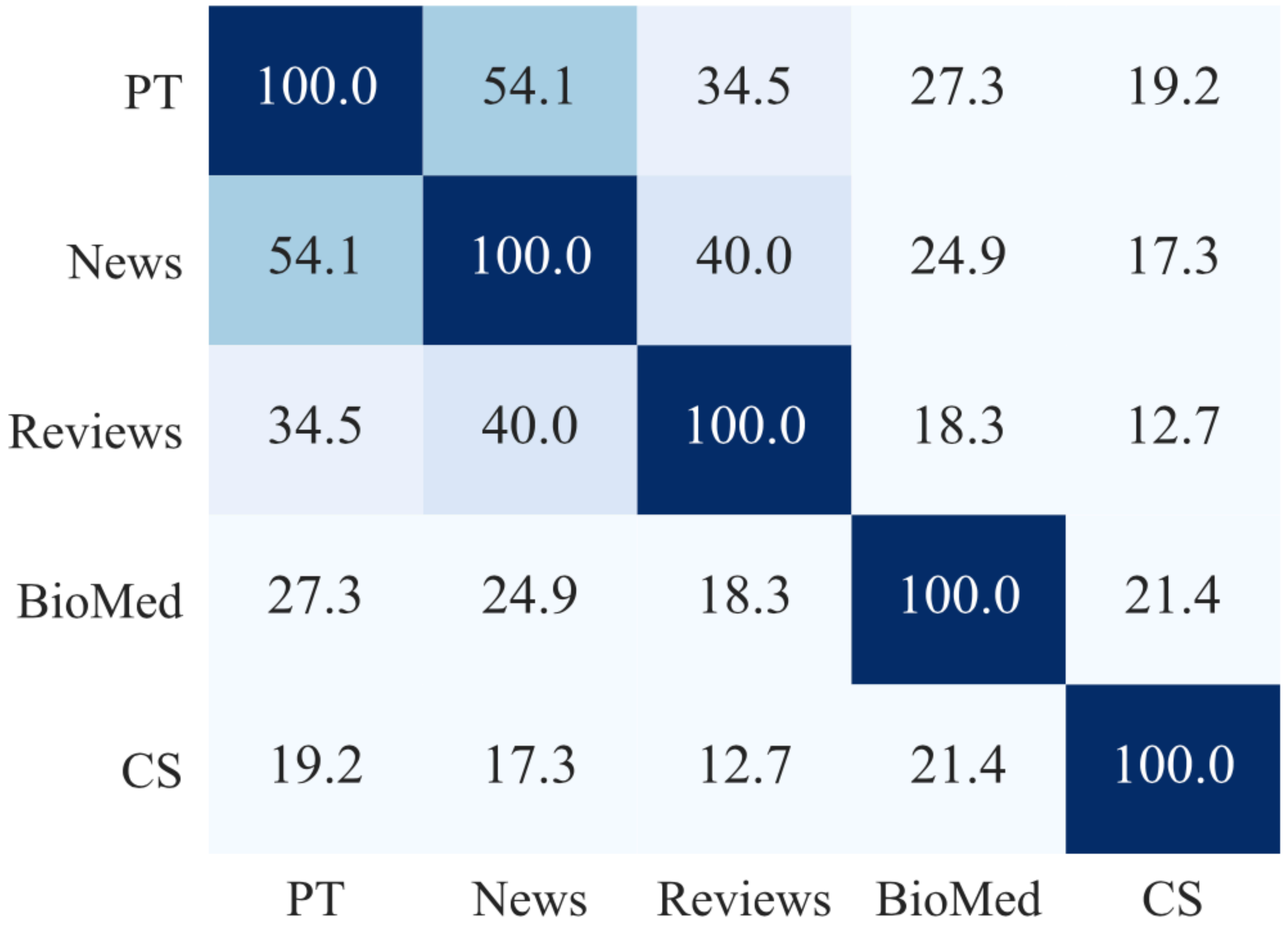
Domain	Pretraining Corpus	# Tokens	Size	$\mathcal{L}_{\text{RoB.}}$	$\mathcal{L}_{\text{DAPT}}$
BioMed	2.68M full-text papers from S2ORC (Lo et al., 2020)	7.55B	47GB	1.32	0.99
CS	2.22M full-text papers from S2ORC (Lo et al., 2020)	8.10B	48GB	1.63	1.34
News	11.90M articles from RealNews (Zellers et al., 2019)	6.66B	39GB	1.08	1.16
Reviews	24.75M Amazon reviews (He and McAuley, 2016)	2.11B	11GB	2.10	1.93
RoBERTa (baseline)	see Appendix §A.1	N/A	160GB	‡1.19	-



Domain	Task	Label Type	Train (Lab.)	Train (Unl.)	Dev.	Test	Classes
BioMed	ChemProt	relation classification	4169	-	2427	3469	13
	†RCT	abstract sent. roles	18040	-	30212	30135	5
CS	ACL-ARC	citation intent	1688	-	114	139	6
	SciERC	relation classification	3219	-	455	974	7
News	HyperPartisan	partisanship	515	5000	65	65	2
	†AGNews	topic	115000	-	5000	7600	4
Reviews	†Helpfulness	review helpfulness	115251	-	5000	25000	2
	†IMDB	review sentiment	20000	50000	5000	25000	2

Paper III - Datasets

- Simple domain corpus similarity calculated by vocabulary overlap. The more dissimilar the domain, the higher the potential for DAPT.

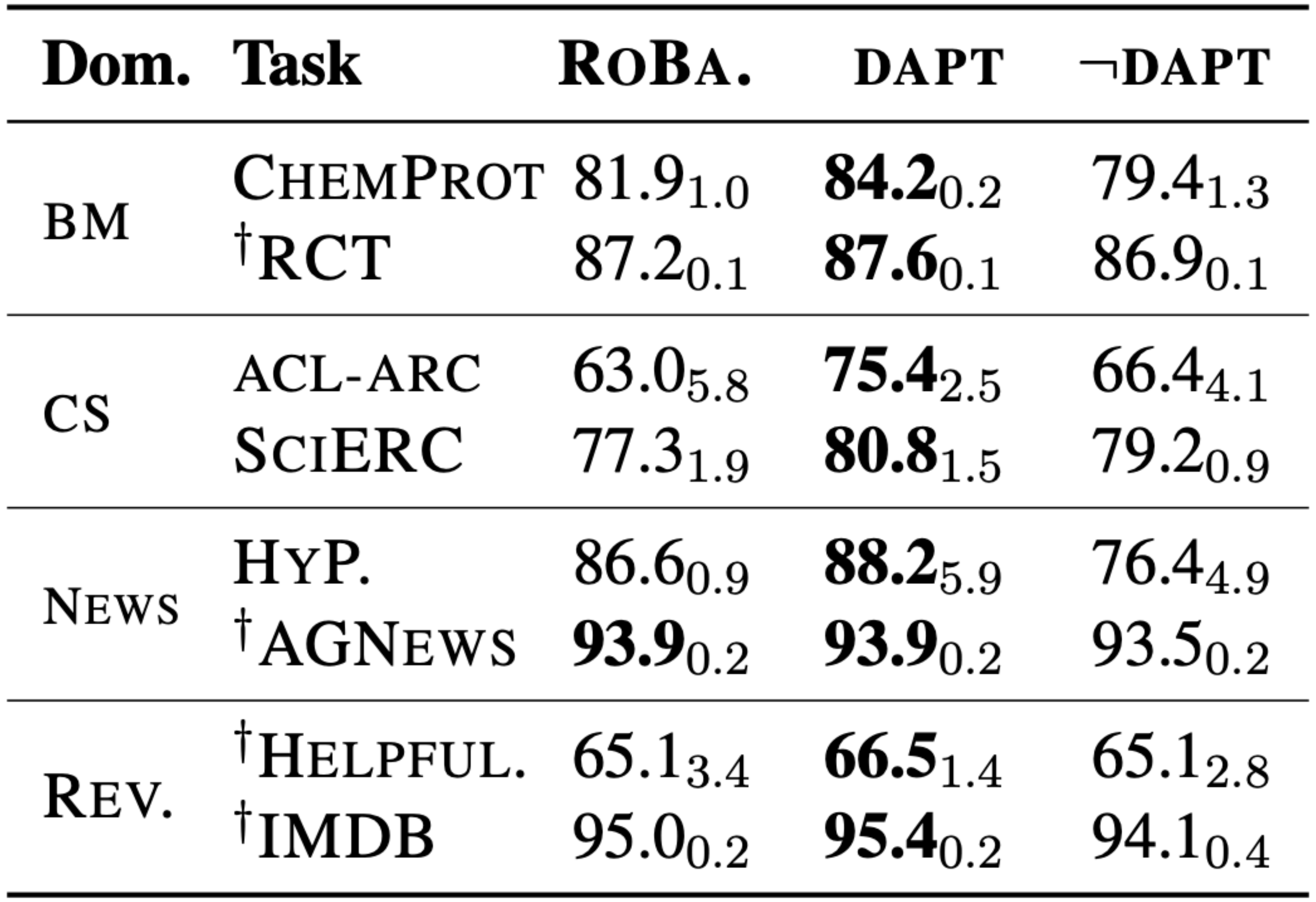


	PT	News	Reviews	BioMed	CS
PT	100.0	54.1	34.5	27.3	19.2
News	54.1	100.0	40.0	24.9	17.3
Reviews	34.5	40.0	100.0	18.3	12.7
BioMed	27.3	24.9	18.3	100.0	21.4
CS	19.2	17.3	12.7	21.4	100.0

Figure 2: Vocabulary overlap (%) between domains. PT denotes a sample from sources similar to RoBERTa's pretraining corpus. Vocabularies for each domain are created by considering the top 10K most frequent words (excluding stopwords) in documents sampled from each domain.

Paper III - Experiment (DAPT)

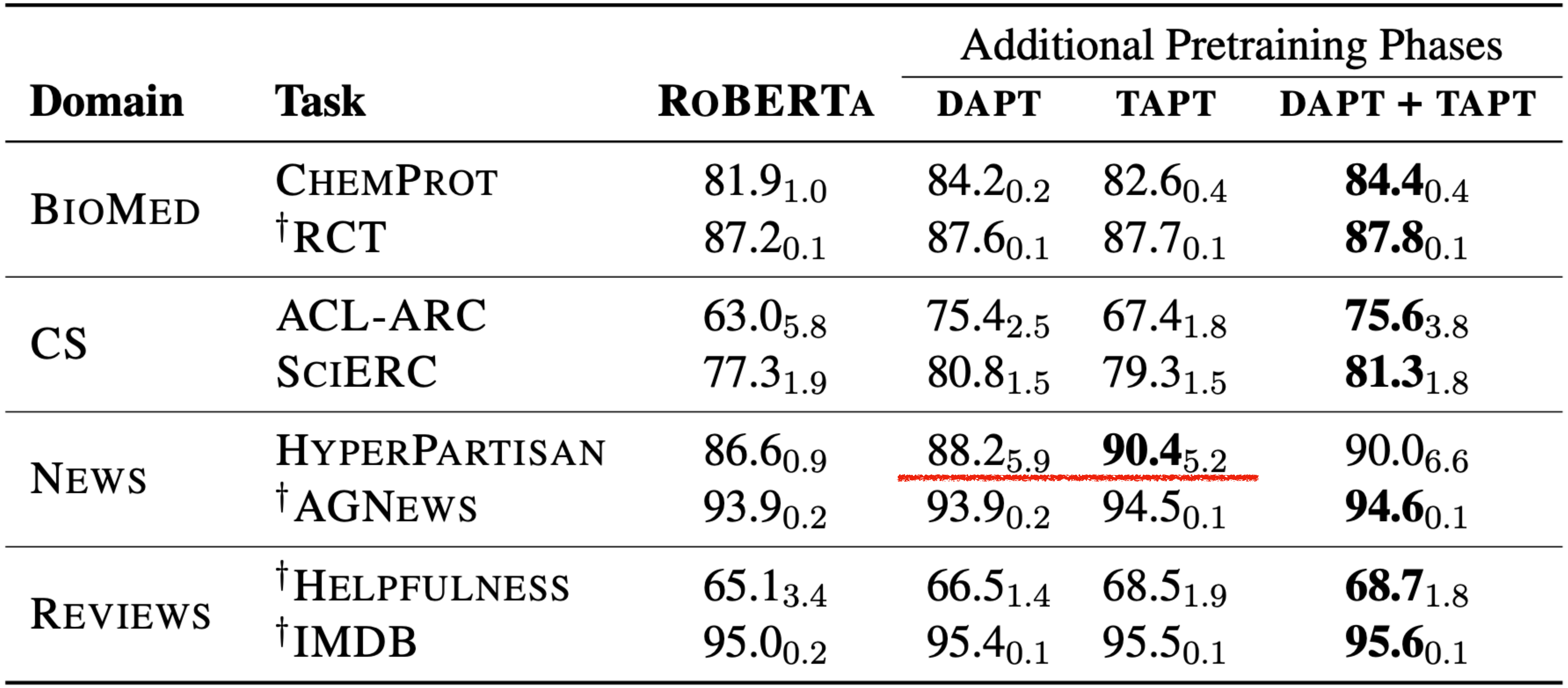
- Experiment setup: 12.5K steps ~= one single pass of each domain corpus.
- Findings:
  - DAPT outperforms RoBERTa.
  - Relevant domain (especially distant domain) benefits RoBERTa more.
  - Adding irrelevant text detriments the performance.



Dom.	**Task**	**RoBa.**	**DAPT**	**¬DAPT**
BM	ChemProt	$81.9_{1.0}$	$\mathbf{84.2}_{0.2}$	$79.4_{1.3}$
	†RCT	$87.2_{0.1}$	$\mathbf{87.6}_{0.1}$	$86.9_{0.1}$
CS	ACL-ARC	$63.0_{5.8}$	$\mathbf{75.4}_{2.5}$	$66.4_{4.1}$
	SciERC	$77.3_{1.9}$	$\mathbf{80.8}_{1.5}$	$79.2_{0.9}$
News	HyP.	$86.6_{0.9}$	$\mathbf{88.2}_{5.9}$	$76.4_{4.9}$
	†AGNews	$\mathbf{93.9}_{0.2}$	$\mathbf{93.9}_{0.2}$	$93.5_{0.2}$
Rev.	†Helpful.	$65.1_{3.4}$	$\mathbf{66.5}_{1.4}$	$65.1_{2.8}$
	†IMDB	$95.0_{0.2}$	$\mathbf{95.4}_{0.2}$	$94.1_{0.4}$

Paper III - Experiment (TAPT)

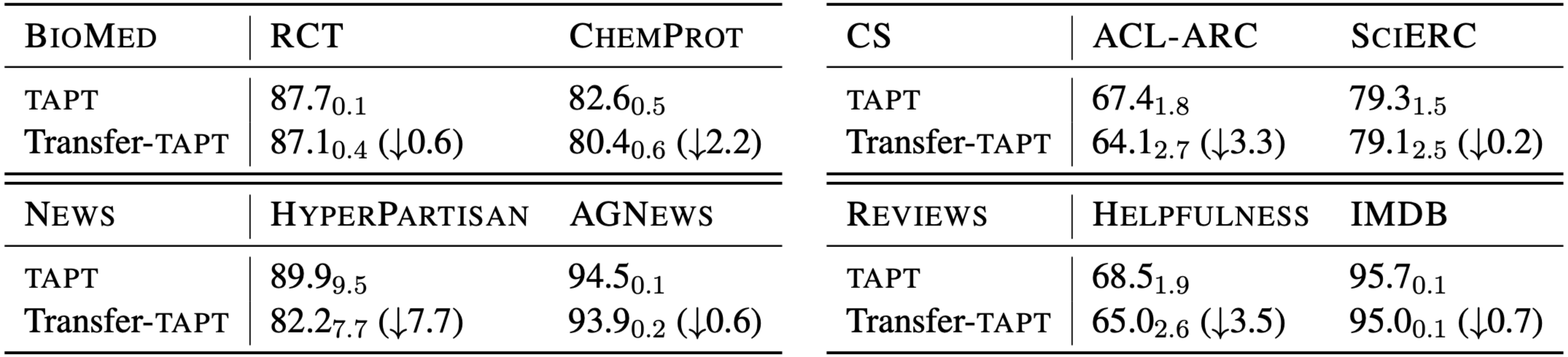
- Experiment setup: 100 epochs
- Findings:
  - TAPT outperforms RoBERTa.
  - TAPT can match and even surpass DAPT which is much more data intensive.
  - DAPT+TAPT yields the best performance.



Domain	Task	RoBERTa	Additional Pretraining Phases		
			DAPT	TAPT	DAPT + TAPT
BioMed	ChemProt	$81.9_{1.0}$	$84.2_{0.2}$	$82.6_{0.4}$	$\mathbf{84.4}_{0.4}$
	†RCT	$87.2_{0.1}$	$87.6_{0.1}$	$87.7_{0.1}$	$\mathbf{87.8}_{0.1}$
CS	ACL-ARC	$63.0_{5.8}$	$75.4_{2.5}$	$67.4_{1.8}$	$\mathbf{75.6}_{3.8}$
	SciERC	$77.3_{1.9}$	$80.8_{1.5}$	$79.3_{1.5}$	$\mathbf{81.3}_{1.8}$
News	HyperPartisan	$86.6_{0.9}$	$88.2_{5.9}$	$\mathbf{90.4}_{5.2}$	$90.0_{6.6}$
	†AGNews	$93.9_{0.2}$	$93.9_{0.2}$	$94.5_{0.1}$	$\mathbf{94.6}_{0.1}$
Reviews	†Helpfulness	$65.1_{3.4}$	$66.5_{1.4}$	$68.5_{1.9}$	$\mathbf{68.7}_{1.8}$
	†IMDB	$95.0_{0.2}$	$95.4_{0.1}$	$95.5_{0.1}$	$\mathbf{95.6}_{0.1}$

Paper III - Experiment (TAPT)

- Further compare DAPT and TAPT by exploring whether adapting to one task transfers to other tasks in the same domain.
- For example, pretrained on RCT and fine-tuning on CHEMPROT.
- The results demonstrate that data distributions of tasks within a given domain might differ.



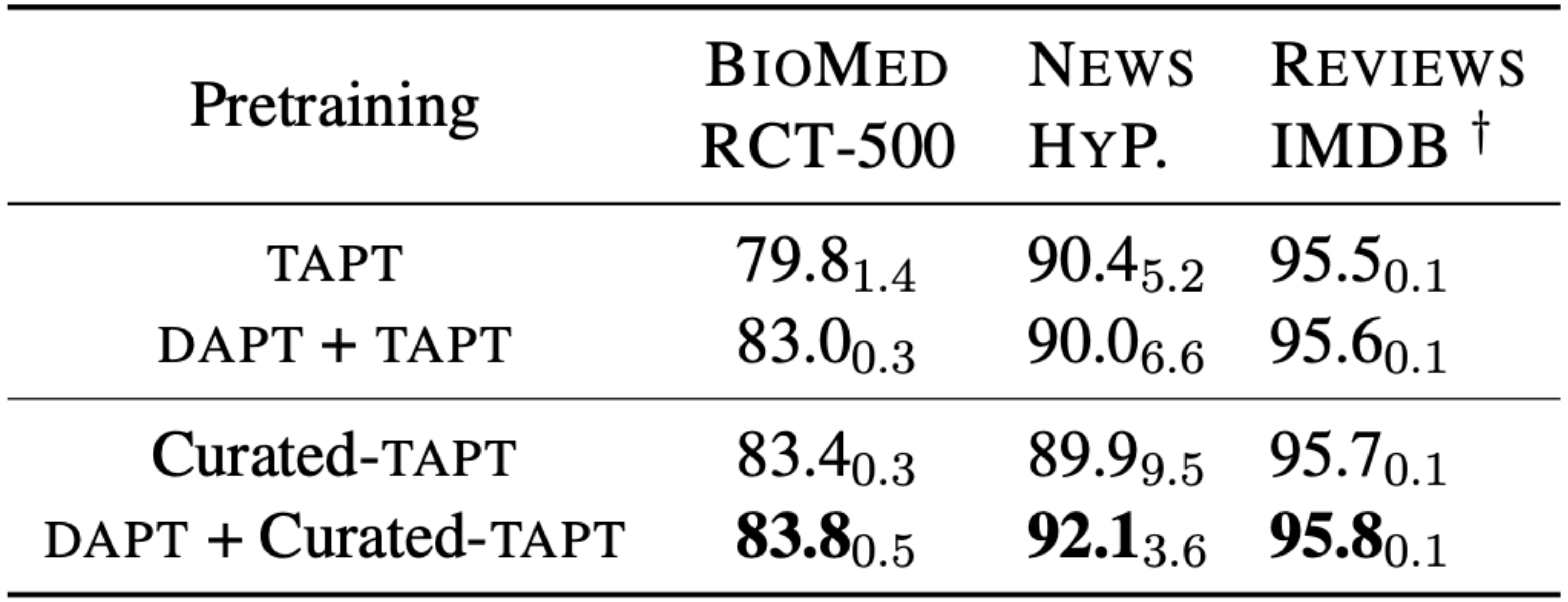
BIOMED	RCT	CHEMPROT
TAPT	$87.7_{0.1}$	$82.6_{0.5}$
Transfer-TAPT	$87.1_{0.4}$ (↓0.6)	$80.4_{0.6}$ (↓2.2)
NEWS	HYPERPARTISAN	AGNEWS
TAPT	$89.9_{9.5}$	$94.5_{0.1}$
Transfer-TAPT	$82.2_{7.7}$ (↓7.7)	$93.9_{0.2}$ (↓0.6)

CS	ACL-ARC	SCIERC
TAPT	$67.4_{1.8}$	$79.3_{1.5}$
Transfer-TAPT	$64.1_{2.7}$ (↓3.3)	$79.1_{2.5}$ (↓0.2)
REVIEWS	HELPFULNESS	IMDB
TAPT	$68.5_{1.9}$	$95.7_{0.1}$
Transfer-TAPT	$65.0_{2.6}$ (↓3.5)	$95.0_{0.1}$ (↓0.7)

Table 6: Though TAPT is effective (Table 5), it is harmful when applied *across* tasks. These findings illustrate differences in task distributions within a domain.

Paper III - Experiment (Augment TAPT)

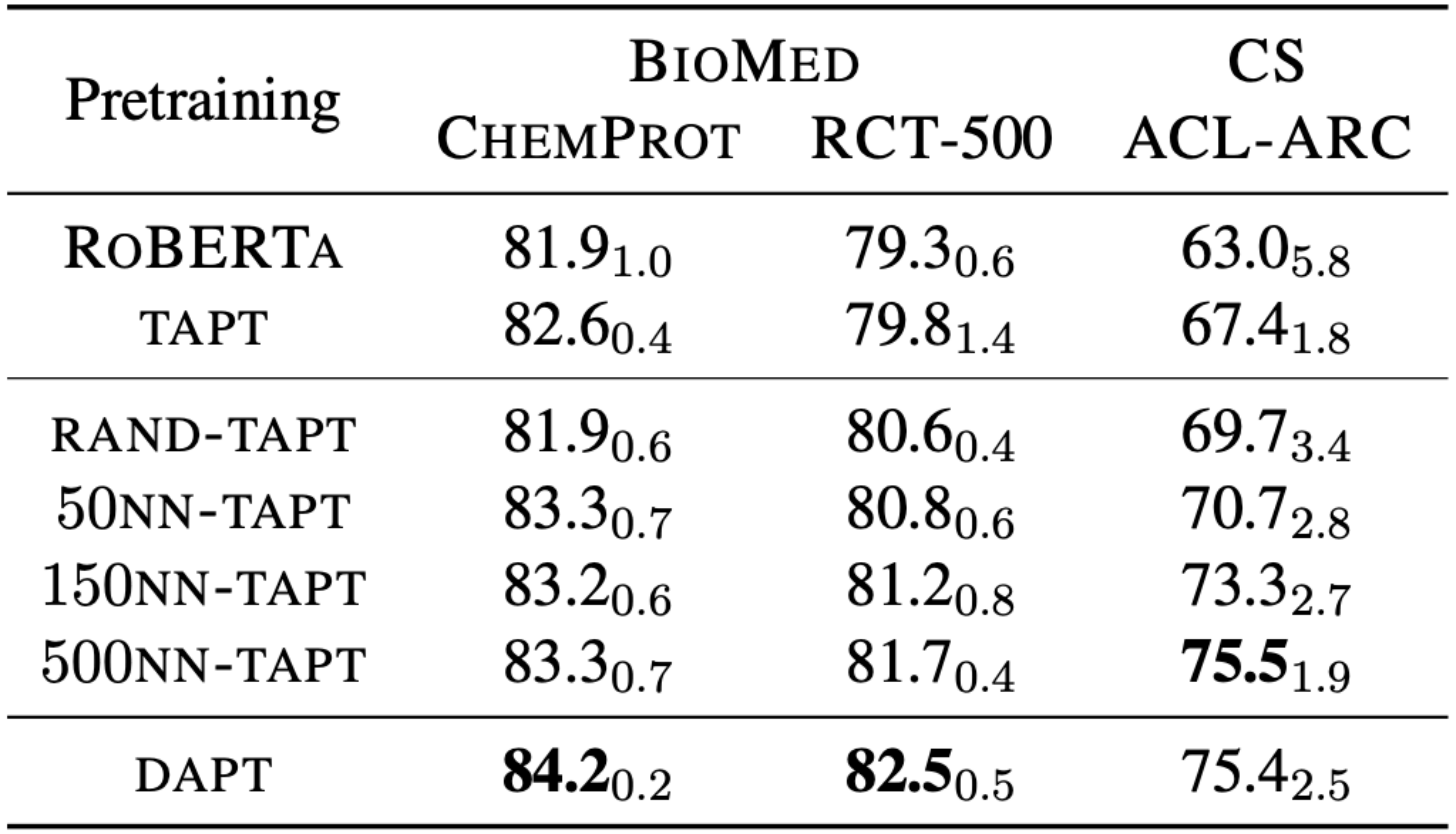
- Inspired by the success of TAPT, the paper augments the training data of TAPT.
- **Augmentation 1 - Human curated TAPT**: larger dataset already exists.
- Data: RCT-500 (downsampling), The HYPERPARTISAN, IMDB.
- Findings:
  - Curated-TAPT matches with DAPT+TAPT. (RCT-500 is only 0.3% of RCT)
  - DAPT+Curated-TAPT achieves the best performance.
  - Curating large amount of data from task distribution is extremely beneficial.



Pretraining	BIOMED RCT-500	NEWS HYP.	REVIEWS IMDB †
TAPT	$79.8_{1.4}$	$90.4_{5.2}$	$95.5_{0.1}$
DAPT + TAPT	$83.0_{0.3}$	$90.0_{6.6}$	$95.6_{0.1}$
Curated-TAPT	$83.4_{0.3}$	$89.9_{9.5}$	$95.7_{0.1}$
DAPT + Curated-TAPT	$\mathbf{83.8}_{0.5}$	$\mathbf{92.1}_{3.6}$	$\mathbf{95.8}_{0.1}$

Paper III - Experiment (Augment TAPT)

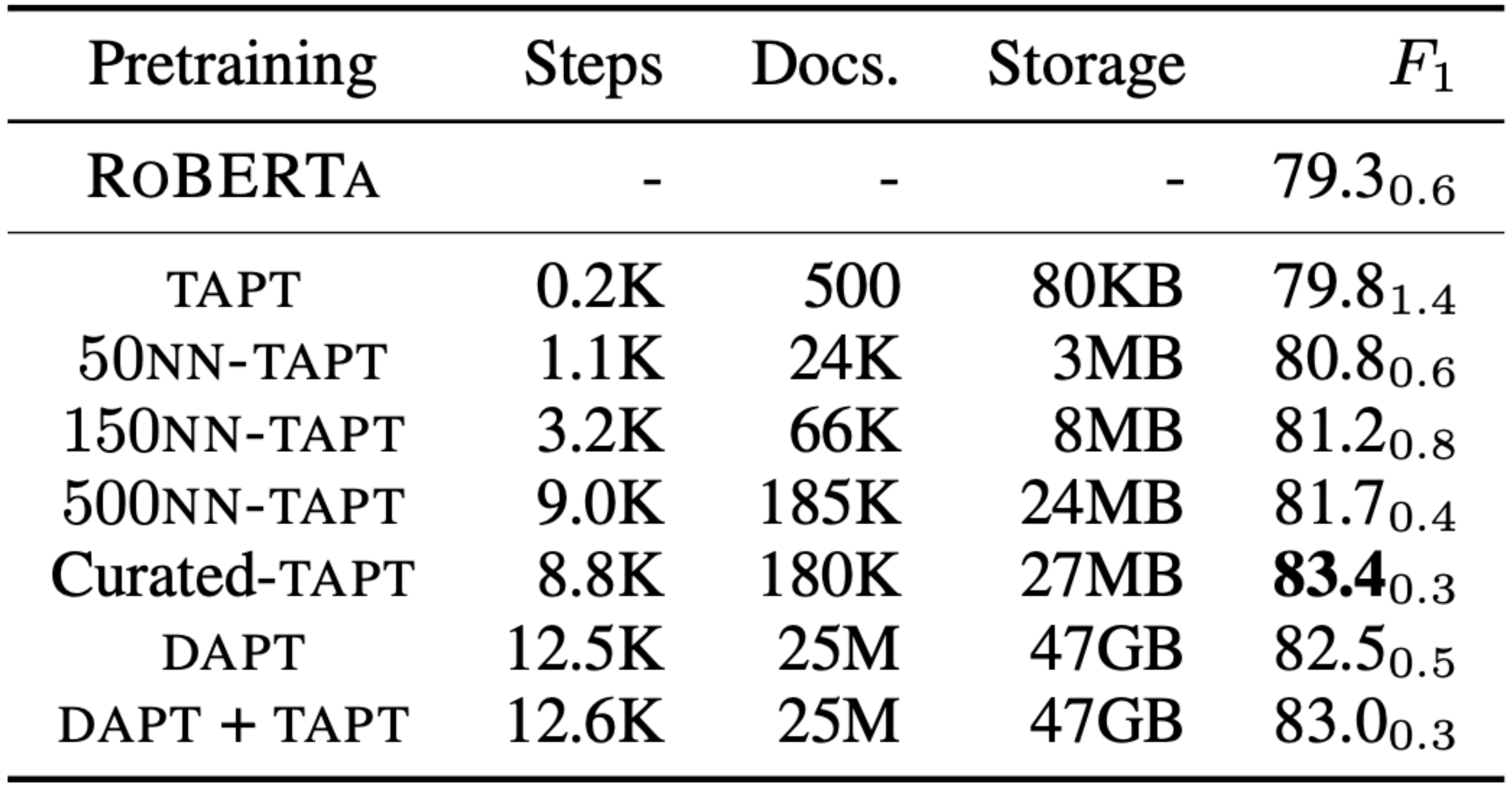
- **Augmentation 2 - Automated data selection for TAPT**: employ VAMPIRE lightweight bag-of-word model to embed both task and domain text, then use KNN on the shared embedding space to collect augmented data.
- Findings:
  - Better than just TAPT, and close to DAPT.
  - Need for better data selection techniques to improve KNN-TAPT.



Pretraining	BioMed		CS
	ChemProt	RCT-500	ACL-ARC
RoBERTa	$81.9_{1.0}$	$79.3_{0.6}$	$63.0_{5.8}$
TAPT	$82.6_{0.4}$	$79.8_{1.4}$	$67.4_{1.8}$
RAND-TAPT	$81.9_{0.6}$	$80.6_{0.4}$	$69.7_{3.4}$
50NN-TAPT	$83.3_{0.7}$	$80.8_{0.6}$	$70.7_{2.8}$
150NN-TAPT	$83.2_{0.6}$	$81.2_{0.8}$	$73.3_{2.7}$
500NN-TAPT	$83.3_{0.7}$	$81.7_{0.4}$	$\mathbf{75.5}_{1.9}$
DAPT	$\mathbf{84.2}_{0.2}$	$\mathbf{82.5}_{0.5}$	$75.4_{2.5}$

Paper III - Experiment

- Computational requirements for different methods:
- TAPT is nearly 60 times faster to train than DAPT on a single v3-8 TPU and storage requirements for DAPT on this task are 5.8M times that of TAPT.



Pretraining	Steps	Docs.	Storage	F_1
RoBERTa	-	-	-	79.3$_{0.6}$
TAPT	0.2K	500	80KB	79.8$_{1.4}$
50NN-TAPT	1.1K	24K	3MB	80.8$_{0.6}$
150NN-TAPT	3.2K	66K	8MB	81.2$_{0.8}$
500NN-TAPT	9.0K	185K	24MB	81.7$_{0.4}$
Curated-TAPT	8.8K	180K	27MB	**83.4**$_{0.3}$
DAPT	12.5K	25M	47GB	82.5$_{0.5}$
DAPT + TAPT	12.6K	25M	47GB	83.0$_{0.3}$

Table 9: Computational requirements for adapting to the RCT-500 task, comparing DAPT (§3) and the various TAPT modifications described in §4 and §5.

Paper III - Conclusion

- Na thorough analysis of domain- and task- adaptive pretraining across four domains and eight tasks, spanning low- and high-resource settings;
- An investigation into the transferability of adapted LMs across domains and tasks.
- A study highlighting the importance of pretraining on human-curated datasets, and a simple data selection strategy to automatically approach this performance.