A APPENDIX

A.1 DATASET DETAILS

A.1.1 LICENCE

We use three publicly-available datasets to construct our benchmarks. These datasets can be downloaded from their original hosts under their terms and conditions:

- FUNSD Jaume et al. (2019) License, instructions to download, and term of use can be found at https://guillaumejaume.github.io/FUNSD/work/
- SROIE Huang et al. (2019) License, instructions to download, and term of use can be found at https://github.com/zzzDavid/ICDAR-2019-SROIE
- DocVQA Mathew et al. (2021) License, instructions to download, and term of use can be found at https://www.docvqa.org/datasets/doccvqa

A.1.2 DATASET SPLITS

We provide the list of the documents in the *source* and *target* domains for our three benchmarks. Files are located at `Supplemental/TTA_Benchmarks/`. For FUNSD-TTA and SROIE-TTA, the validation splits have 10 and 39 documents, respectively, which are selected randomly using the seed number 42. Validation splits have a similar distribution as the source domain's training data. When performing TTA, we use the target domain data without labels – the labels are only used for evaluation purposes. Table 5 and Table 6 show the statistics of documents on source and target domains in FUNSD-TTA and SROIE-TTA, respectively.

Table 4: Number of documents in the source and target domains in FUNSD-TTA and SROIE-TTA benchmarks. We use the validation set selected from the source domain to tune TTA algorithm's hyper parameters.

Table 5: FUNSD-TTA

Source Training	139
Source Validation	10
Source Evaluation, Target Training, Target Evaluation	50

Table 6: SROIE-TTA

Source Training	600
Source Validation	39
Source Evaluation, Target Training, Target Evaluation	347

For DocVQA-TTA benchmark, we always choose 10% of source domain data for validation using the same seed (42).

Table 7: Number of documents in each domain of our DocVQA-TTA benchmark.

	Layout (**L**)	Emails&Letters (**E**)	Tables&Lists (**T**)	Figures&Diagrams (**F**)
Source Training	1807	1417	592	150
Source Validation	200	157	65	17
Source Evaluation, Target Training, Target Evaluation	512	137	187	49

A.1.3 TEXT EMBEDDINGS AND OCR ANNOTATIONS

For all the benchmarks, we use officially-provided OCR annotations for each datasets. For the tokenization process, we follow Xu et al. (2020a) where they used WordPiece Wu et al. (2016) such that each token in the OCR text sequence is assigned to a certain segment of $s_i \in \{[\texttt{A}], [\texttt{B}]\}$ prepended by [CLS] if it is the starting token and/or appended by [SEP] if it is the ending token of the sequence. In order to have a fixed sequence length in each document, extra [PAD] tokens are appended to the end, if the sequence exceeds a maximum length threshold (512 in this work).

A.2 EXPERIMENTS DETAILS

A.2.1 TRAINING.

We use PyTorch (Paszke et al., 2019) on Nvidia Tesla V100 GPUS for all the experiments. For **source training**, we use LayoutLMv2$_{BASE}$ pre-trained on IIT-CDIP dataset and fine-tune it with labeled source data on our desired task. For all VDU tasks, we build task-specific classifier head layers over the text embedding of LayoutLMv2$_{BASE}$ outputs. For entity recognition and key-value extraction tasks, we use the standard cross-entropy loss and for DocVQA task, we use the binary cross-entropy loss on each token to predict whether it is the starting/ending position of the answer or not. We use AdamW (Loshchilov & Hutter, 2017) optimizer and train source model with batch sizes of 32, 32, and 64 for 200, 200, and 70 epochs with a learning rate of 5×10^{-5} for entity recognition, key-value extraction, and DocVQA benchmarks, respectively with an exception of *Figures & Diagrams* domain on which we used a learning rate of 10^{-5}. For BN and SHOT baselines, we followed SHOT implementation for image classification and added a fully connected layer with 768 hidden units, followed by a batch normalization layer right before the classification head.

Uncertainty and confidence-aware pseudo labeling For uncertainty-aware pseudo labeling, we set a threshold (γ) above which pseudo labels are rejected to be used for training. Likewise, for confidence-aware pseudo labeling, we set a threshold for the output probability values for the predicted class *below* which pseudo labels are rejected. For the combination of the two, a pseudo label which has confidence (output probability) value above the threshold and uncertainty value (Shannon entropy) below the maximum threshold is chosen for self-training. We used confidence threshold of 0.95 and tuned the uncertainty threshold to be either 1.5 or 2 (see below).

A.2.2 HYPER PARAMETER TUNING

We used a validation set (from source domain) in each benchmark for hyper parameter tuning. Although not optimal, it is more realistic to assume no access to any labeled data in the target domain. We used a simple grid search to find the optimal set of hyper parameters with the following search space:

- Learning rate $\in \{10^{-5}, 2.5 \times 10^{-5}, 5 \times 10^{-5}\}$
- Weight decay $\in \{0, 0.01\}$
- Batch size $\in \{1, 4, 5, 8, 32, 40, 48, 64\}$
- Uncertainty threshold $\gamma \in \{1.5, 2\}$

A.3 MEASURING CONFIDENCE AFTER ADAPTATION WITH DOCTTA

For reliable VDU deployments, confidence calibration can be very important, as it is desired to identify when the trained model can be trusted so that when it is not confident, a human can be consulted. In this section, we focus on model calibration and analyze how DocTTA affects it. Figure 3 illustrates reliability diagrams for adapting from Emails & Letters (**E**) to **T**, **F**, and **L** domains in DocVQA-TTA benchmark. We compares model calibration before and after DocTTA for 'starting position index of an extracted answer' in documents. We illustrate calibration with reliability diagram (DeGroot & Fienberg, 1983; Niculescu-Mizil & Caruana, 2005), confidence histograms (Guo et al., 2017), and the ECE metric. The reliability diagram shows the expected accuracy as a function of confidence. We first group the predictions on target domain into a set of bins (we used 10). For each bin, we then compute the average confidence and accuracy and visualize them (top red plots in Fig. 3). The closer the bars are to the diagonal line, the more calibrated the model would be. Also, lower ECE values

Table 8: Standard deviations (in parentheses) for ANLS scores shown in Table 2 of the main paper for adapting between domains in **DocVQA-TTA** benchmark (Part I).

Source:	Emails&Letters (**E**)			Figures&Diagrams (**F**)		
Target:	**F**	**T**	**L**	**E**	**T**	**L**
Source-only	37.79 (1.30)	25.59 (1.78)	38.25 (0.92)	5.23 (2.86)	7.03 (2.42)	3.65 (2.76)
DANN	38.94 (1.20)	27.22 (1.32)	40.23 (1.78)	15.43 (1.34)	9.34 (3.23)	7.45 (3.29)
CDAN	39.08 (1.59)	29.33 (0.82)	41.29 (2.80)	16.99 (1.56)	11.32 (2.43)	10.23 (2.54)
DocUDA (ours)	**39.23** (1.42)	**43.54** (0.91)	**57.99** (0.12)	**24.21** (0.72)	**15.76** (0.67)	**20.45** (0.43)
BN	38.10 (1.01)	26.89 (0.59)	38.23 (0.98)	7.32 (2.43)	8.56 (2.34)	9.35 (3.21)
TENT	38.34 (0.74)	26.42 (0.52)	40.45 (0.81)	12.38 (3.12)	7.34 (2.54)	11.29 (2.45)
SHOT	38.98 (0.89)	27.55 (0.81)	39.15 (1.23)	14.34 (3.87)	10.10 (1.34)	13.21 (2.54)
DocTTA (ours)	**40.36** (0.53)	**35.28** (0.76)	**49.35**(1.20)	**22.91**(0.45)	**15.67**(0.78)	**16.01**(1.18)
Train-on-target	95.28 (1.32)	93.54 (0.91)	95.01 (1.34)	39.70 (1.02)	24.77 (0.23)	38.59 (0.78)

indicate better calibrations. It is observed that calibration improves with DocTTA. From this plot, we can also measure ECE as a summary metric (the lower, the better calibration). For instance, DocTTA on **E** $\rightarrow$ **L** yields significantly lower ECE, from 30.45 to 2.44. Although the reliability diagram can explain model's calibration well, it does not show the portion of samples at a given bin. Thus, we use confidence histograms (see bottom of Fig. 3) where the gap between accuracy and average confidence is indicative of calibration. Before adaptation, the model tends to be overconfident whereas, after adaptation with DocTTA, the gap becomes drastically smaller and nearly overlaps.

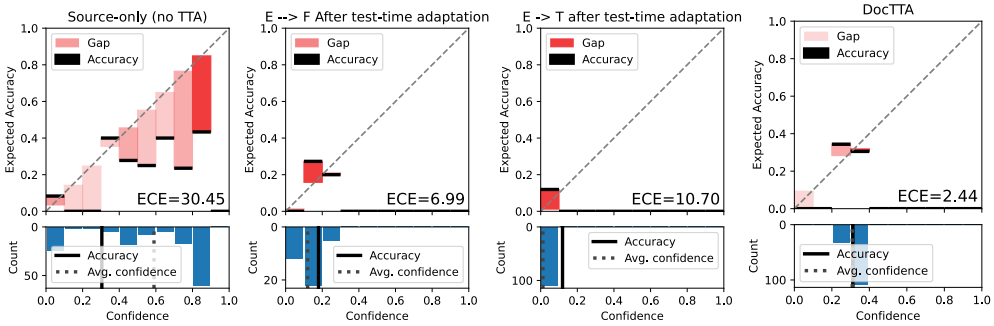

Figure 3: Comparing confidence calibration with and without DocTTA when adapting from Emails & Letters domain to other domains in DocVQA-TTA benchmark.

A.4 Standard deviations

Here we show the standard deviations obtained over 3 seeds for results reported in Table 2 of the main paper in Table 8 (part I) and 9 (part II), respectively.

A.5 DocUDA algorithm

In DocUDA, we use source data during training on target at test time. Therefore, DocUDA has an additional objective function which is a cross-entropy loss using labeled source data:

$$\mathcal{L}_{CE_{Src}}(\theta_t) = -\mathbb{E}_{x_s \in \mathcal{X}_s} \sum_{c=1}^{C} y_c \log \sigma(f_t(x_s)), \tag{6}$$

And the overall objective function of DocUDA is Eq. 5 plus Eq. 6:

$$\mathcal{L}_{\text{DocUDA}} = \mathcal{L}_{MVLM} + \mathcal{L}_{CE} + \mathcal{L}_{DIV} + \mathcal{L}_{CE_{Src}}. \tag{7}$$

Algorithm 2 shows how DocUDA is performed on VDU tasks.

Table 9: Standard deviations (in parentheses) for ANLS scores shown in Table 2 of the main paper for adapting between domains in **DocVQA-TTA** benchmark (Part II).

Source:	Tables&Lists (**T**)			Layout (**L**)		
Target:	**E**	**F**	**L**	**E**	**F**	**T**
Source-only	13.66 (1.67)	20.48 (1.54)	14.58 (1.30)	53.55 (1.76)	33.36 (1.84)	33.43 (1.94)
DANN	17.67 (1.49)	22.19 (2.34)	17.67 (2.58)	54.55 (0.72)	33.87 (1.02)	33.58 (0.28)
CDAN	27.87 (1.82)	25.23 (1.24)	27.66 (2.62)	56.82 (0.62)	34.27 (0.82)	34.81 (0.43)
DocUDA (ours)	**53.19** (0.92)	**29.91** (0.45)	**47.81**(0.41)	**61.09** (0.08)	**34.85** (0.16)	**41.80** (0.06)
BN	15.13 (2.04)	22.24 (1.29)	15.65 (2.43)	53.23 (1.08)	33.67 (1.17)	33.55 (1.55)
TENT	16.01 (2.83)	20.23 (2.61)	15.02 (2.83)	53.34 (0.93)	33.59 (1.82)	34.55 (1.44)
SHOT	22.56 (1.12)	24.33 (2.54)	19.15 (2.39)	56.23 (1.20)	34.56 (1.02)	35.65 (1.23)
DocTTA (ours)	**35.67** (1.10)	**30.70** (0.80)	**26.32** (1.27)	**59.84** (0.04)	**37.01** (0.05)	**39.10** (0.06)
Train-on-target	84.59 (1.02)	70.66 (0.82)	83.73 (0.23)	92.32 (0.01)	91.36 (0.04)	93.41 (0.05)

Algorithm 2 DocUDA for closed-set UDA in VDU

1: **Input:** labeled source documents $\{x_s^{(i)}, y_s^{(i)}\}_{i=1}^{n_s}$, target documents $\{x_t^i\}_{i=1}^{n_t}$, test-time training epochs n_e, test-time training learning rate α, uncertainty threshold γ
2: **Initialization:** Initialize target model f_{θ_t} with LayoutLMv2$_{BASE}$ weights trained on IIT-CDIP dataset.
3: **for** $epoch = 1$ to n_e **do**
4: Perform masked visual-language modeling in Eq. 1
5: Generate pseudo labels and accept a subset using criteria in Eq. 3 and fine-tune with Eq. 2
6: Maximize diversity in pseudo label predictions Eq. 4
7: Perform supervised training using labeled source data with Eq. 2
8: $\theta_t \leftarrow \theta_t - \alpha \nabla \mathcal{L}_{\text{DocUDA}}$ ▷ Update θ_t via total loss in Eq. 7
9: **end for**

A.6 Qualitative Results

Here we show a randomly selected document from our FUNSD-TTA benchmark. Comparing the results before and after using DocTTA shows that our method can refine some wrong predictions made by the unadapted model.

Figure 4: From left to right we show predictions made by (i) an unadapted model, (ii) after using DocTTA, iii) ground truth labels.

A.7 ADDITIONAL BASELINE RESULTS

Here we show the performance of our model against AdaContrast (Chen et al., 2022) for TTA and SHOT-UDA (Liang et al., 2020) on DocVQA-TTA benchmark when adapting from *Emails & Letters* domain to **F**, **T**, and **L**.

Table 10

Source	Emails&Letters (E)		
Target	**F**	**T**	**L**
SHOT (UDA)	39.02	31.35	48.87
DocUDA (ours)	**39.23**	**43.54**	**57.99**
AdaContrast (TTA)	37.21	27.43	38.69
DocTTA (ours)	**40.36**	**35.28**	**49.35**

A.8 LAYOUT ILLUSTRATION

In our method, layout refers to a bounding box associated with each word in the text input sequence and is represented with a 6-dimensional vector in the form of $(x_{min}, x_{max}, y_{min}, y_{max}, w, h)$. where (x_{min}, y_{min}) corresponds to the position of the lower left corner and (x_{max}, y_{max}) represents the position of the upper right corner of the bounding box and w and h denote the width and height of the box, respectively as shown below

Figure 5: Illustration of layout as a bounding box associated with each word in the text input sequence.