

Document Image Machine Translation with Dynamic Multi-pre-trained Models Assembling

Yupu Liang^{1,2}, Yaping Zhang^{1,2*}, Cong Ma^{1,2}, Zhiyang Zhang^{1,2},
Yang Zhao^{1,2}, Lu Xiang^{1,2}, Chengqing Zong^{1,2}, Yu Zhou^{1,3}

¹ State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS),
Institute of Automation, Chinese Academy of Sciences, Beijing, China

² School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

³ Fanyu AI Laboratory, Zhongke Fanyu Technology Co., Ltd, Beijing, China

{liangyupu2021, zhangzhiyang2020}@ia.ac.cn, {yaping.zhang, cong.ma, yang.zhao, lu.xiang, cqzong, yzhou}@nlpr.ia.ac.cn

Abstract

Text image machine translation (TIMT) is a task that translates source texts embedded in the image to target translations. The existing TIMT task mainly focuses on text-line-level images. In this paper, we extend the current TIMT task and propose a novel task, **Document Image Machine Translation to Markdown (DIMIT2Markdown)**, which aims to translate a source document image with long context and complex layout structure to markdown-formatted target translation. We also introduce a novel framework, **Document Image Machine Translation with Dynamic multi-pre-trained models Assembling (DIMITDA)**. A dynamic model assembler is used to integrate multiple pre-trained models to enhance the model’s understanding of layout and translation capabilities. Moreover, we build a novel large-scale **Document image machine Translation dataset of ArXiv articles in markdown format (DoTA)**, containing 126K image-translation pairs. Extensive experiments demonstrate the feasibility of end-to-end translation of rich-text document images and the effectiveness of DIMITDA.¹

1 Introduction

Text Image Machine Translation (TIMT) is an emerging field focused on translating text from one language to another within images, as explored by Lan et al. (2023). Recent studies in TIMT fall into two primary categories: (1) Cascade systems (Sable et al., 2023; Lan et al., 2023; Zhang et al., 2023), which involve sequential multi-model operations. These systems often grapple with structural redundancy, error propagation, and high latency. (2) End-to-end models (Jain et al., 2021; Ma et al., 2022; Zhu et al., 2023), optimizing the entire model through a unified training objective with more efficient structure.

*Corresponding author.

¹Our dataset and code are available at: <https://github.com/liangyupu/DIMITDA>





Image Type	Input Image	Output Text	Output Format
Scene Text Image		“Industrial & Commercial Bank of China”	Plain Text
Text Line Image		“Scientists working in medicine”	Plain Text
Document Image			Markdown Text (after rendering)

Figure 1: The illustration of different text image machine translation tasks.

	Text-line-level Images	Document Images
ItNet (Jain et al., 2021)	39.30	3.84
E2ETIT (Ma et al., 2022)	15.69	1.51
PEIT (Zhu et al., 2023)	47.20	5.81

Table 1: BLEU scores of end-to-end TIMT models on different image scenarios. All methods achieve commendable results on text-line-level images but fail on document images.

Existing end-to-end methods (Jain et al., 2021; Ma et al., 2022; Zhu et al., 2023) have shown promising results on the TIMT task. However, as shown in Figure 1, these methods are primarily tailored for text-line-level applications with short texts, such as road signs, shop billboards, and subtitles. But document images are also prevalent in real-world scenarios, such as academic papers, magazines, and scanned documents. Unfortunately, direct application of these end-to-end TIMT methods to document images with long context and complex layout encounters significant challenges. As shown in Table 1², we verify the applicability of these on document images, including ItNet (Jain et al., 2021), E2ETIT (Ma et al., 2022) and PEIT

²For text-line-level images, we directly report the results from their corresponding papers. For document images, we train the models on our DoTA dataset.

(Zhu et al., 2023). We observe that while these end-to-end TIMT methods exhibit satisfactory performance on text-line-level images, they all struggle to handle document images. For example, the best model, PEIT, achieves a BLEU score of 47.20 for text-line-level images, while only achieving a BLEU score of 5.81 for document images.

We think there are three reasons for the poor performance of the current TIMT models on document images: (1) **Long context**: the number of tokens contained in text-line-level images is typically less than 50, whereas the number of tokens in document images ranges from several hundred to over a thousand; (2) **Complex layout**: document images contain complex layouts, like title, paragraph, table, figure, and formula, which severely hinder the performance of the model; (3) **Data scarcity**: there is no large-scale public dataset available for the end-to-end Document Image Machine Translation (DIMIT) task. These three reasons lead to the slow convergence and poor performance of the end-to-end model.³

To address the above issues, we extend the current end-to-end TIMT task and introduce a novel **Document Image Machine Translation to Markdown (DIMIT2Markdown)** task. This novel task aims to translate document images from one language to another while meticulously preserving the logical layout in markdown format, as shown in Figure 1.

Moreover, to alleviate the convergence problem of the end-to-end model, we propose a novel framework for **Document Image Machine Translation with Dynamic multi-pre-trained models Assembling (DIMITDA)**, which leverages the multiple pre-trained models to initialize the end-to-end DIMIT model. DIMITDA is a pure end-to-end framework that contains a model assembler to dynamically connect multi-pre-trained models. Specifically, it contains four components: (1) a pre-trained optical character recognition (OCR) model to capture the textual information; (2) a pre-trained layout model to encode the layout structure information; (3) a pre-trained translation decoder to generate translated texts; (4) a dynamic model assembler to fuse the pre-trained models.

Besides, to alleviate data scarcity, we construct a large-scale dataset for the DIMIT2Markdown task, **DoTA**, containing 126K scientific document images collected from arXiv paired with Chinese

translation texts in markdown format.

Our main contributions are concluded as follows:

- We introduce a novel image translation task, DIMIT2Markdown, translating a source document image into target text in markdown format, which expands the research scope in text image machine translation.
- We propose a novel end-to-end framework, DIMITDA, that can dynamically assemble multi-pre-trained models to complete the DIMIT2Markdown task.
- We build a large-scale dataset, DoTA, to facilitate the training and evaluation of the DIMIT2Markdown task.

2 Related Work

Text image machine translation is to translate the source text image to target translations. In recent years, various end-to-end methods (Mansimov et al., 2020; Jain et al., 2021; Ma et al., 2022, 2023a,b,c; Zhu et al., 2023) have been proposed. Jain et al. (2021) uses a convolutional encoder and an autoregressive Transformer decoder to build the model. Ma et al. (2022) trains the end-to-end TIMT model with text translation as an auxiliary task. Zhu et al. (2023) proposes an end-to-end TIMT framework that bridges the modality gap with pre-trained models. While these end-to-end methods have demonstrated satisfactory performance, their effectiveness is limited to images with short context and simple layout structure.

Multi-model machine translation aims to improve text machine translation performance by incorporating visual information (Huang et al., 2021). There have been many studies (Yao and Wan, 2020; Caglayan et al., 2021; Li et al., 2022a; Guo et al., 2023) focused on this area of research. The biggest distinction between multi-model machine translation and our DIMIT2Markdown task lies in the input difference. Multi-model machine translation takes source text as input with images as auxiliary, while the only input for our task is a document image.

With the advancement of document image research, numerous pre-trained models (Kim et al., 2022; Bao et al., 2022; Li et al., 2022b, 2023b; Blecher et al., 2023; Lv et al., 2023), especially for document images, have emerged. These models have shown excellent performance in their respective downstream tasks, demonstrating robust capabilities in understanding document images. This

³This can be confirmed by Table 4 line 10 and 11.

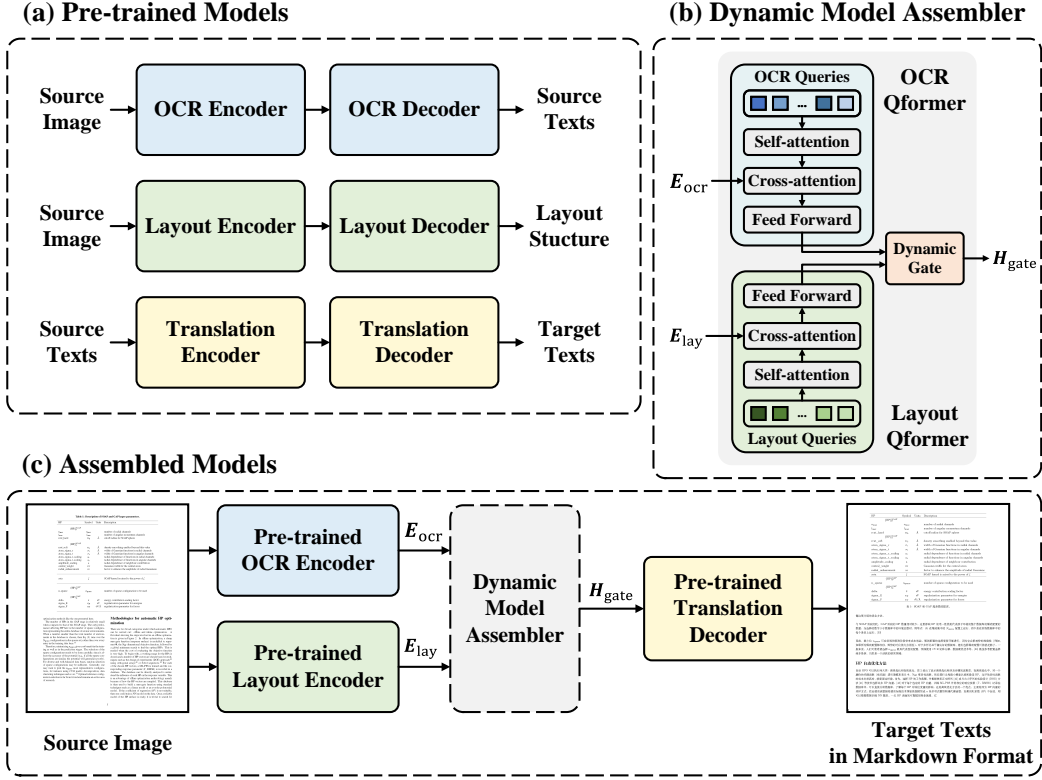


Figure 2: The diagram of the proposed DIMTDA with an OCR encoder, a layout encoder, a dynamic model assembler, and a translation decoder. The OCR encoder, layout encoder, and translation decoder are initialized from pre-trained models.

inspires us to leverage the knowledge of multi-pre-trained models for the DIMT2Markdown system.

3 Method Description

3.1 Task Formulation

The input of the DIMT2Markdown task is a source document image with complex layout structure. The output is target translations in markdown format. The dataset can be denoted as $\mathcal{D} = \{(I, Y)\}$, where I denotes the input image and $Y = \{y_1, y_2, \dots, y_n\}$ denotes the structured target text. The goal of an end-to-end model can be formulated as:

$$\mathcal{L}_{\text{DIMT2Markdown}} = - \sum_{t=1}^n \log p(y_t | y_{<t}, I; \theta) \quad (1)$$

where θ is the parameter of the whole model.

3.2 Model Architecture

The model architecture of DIMTDA is illustrated in Figure 2.

OCR Encoder It encodes the input image I to its semantic representation. We use Swin Transformer (Liu et al., 2021) to construct the OCR en-

coder.⁴ Given the input image $I \in \mathbb{R}^{H \times W \times 3}$, the feature sequence E_{ocr} output by the OCR encoder can be formulated as:

$$E_{\text{ocr}} = \text{Encoder}_{\text{ocr}}(I) \in \mathbb{R}^{l_{\text{ocr}} \times d_{\text{ocr}}} \quad (2)$$

where l_{ocr} and d_{ocr} are the sequence length and dimension of the feature separately.

Layout Encoder This encoder encodes the global structure information into a series of vectors. ViT (Dosovitskiy et al., 2020) is used to construct the layout encoder.⁵ It takes the split image patches and 1D position embeddings as inputs and outputs a series of encoded vectors which can be denoted as E_{lay} . Formally, E_{lay} is calculated as follows:

$$E_{\text{lay}} = \text{FFN}(\text{Encoder}_{\text{lay}}(I)) \in \mathbb{R}^{l_{\text{lay}} \times d_{\text{ocr}}} \quad (3)$$

where l_{lay} is the number of output vectors and an FFN is required to map the dimension to d_{ocr} .

⁴In preliminary experiments, we experimented with CNN-based, ViT-based, and Swin Transformer-based encoders, ultimately selecting the Swin Transformer for the best performance.

⁵In preliminary experiments, we experimented with CNN-based and ViT-based encoders, ultimately selecting the ViT for the best performance.

Dynamic Model Assembler It is the core module of our framework. Two Qformer modules (Li et al., 2023a) are used to unify the feature sequences from encoders to the same length. Each Qformer module is a one-layer Transformer with a self-attention module, a cross-attention module, and a feed-forward network. We also create two set numbers of learnable query tokens $\mathbf{Q} \in \mathbb{R}^{N \times d_{ocr}}$ as input to the two Qformer modules separately, where N is the number of query tokens.

Let \mathbf{H}_{ocr} and \mathbf{H}_{lay} denote the output sequences of OCR Qformer and layout Qformer, respectively. They can be calculated in the following way:

$$\mathbf{H}_{ocr} = \text{Qformer}_{ocr}(\mathbf{Q}_{ocr}, \mathbf{E}_{ocr}) \in \mathbb{R}^{N \times d_{ocr}} \quad (4)$$

$$\mathbf{H}_{lay} = \text{Qformer}_{lay}(\mathbf{Q}_{lay}, \mathbf{E}_{lay}) \in \mathbb{R}^{N \times d_{ocr}} \quad (5)$$

We employ a soft gate mechanism to dynamically merge the two types of information, allowing the model to determine the relative importance of each. Formally, this process can be shown as follows:

$$\mathbf{H}_{gate} = \text{FFN}(\lambda \odot \mathbf{H}_{ocr} + (1 - \lambda) \odot \mathbf{H}_{lay}) \quad (6)$$

$$\lambda = \text{Sigmoid}(W_{ocr}\mathbf{H}_{ocr} + W_{lay}\mathbf{H}_{lay} + b) \quad (7)$$

where \mathbf{H}_{gate} is the fused feature, and W_{ocr}, W_{lay}, b are trainable parameters. An FFN is needed to map the output feature dimension d_{ocr} to the dimension of translation decoder d_{trans} .

Translation Decoder Similar to the vanilla Transformer’s decoder, this decoder receives the fused feature from the dynamic model assembler for cross-attention computing and generates translation texts in an auto-regressive manner. At each decoding timestep t , the translation decoder takes the fused feature \mathbf{H}_{gate} and generated target tokens $y_{<t} = y_1, y_2, \dots, y_{t-1}$ as input and outputs the probability distribution of next target token y_t . This process is defined as follows:

$$p(y_t | y_{<t}, \mathbf{I}; \theta) = \text{Decoder}(y_{<t}, \mathbf{H}_{gate}) \quad (8)$$

where θ denotes the parameters of the whole model.

3.3 Training Strategy

As shown in Figure 2, the OCR encoder, layout encoder, and translation decoder are initialized from pre-trained models, while the dynamic model assembler is randomly initialized. Then, we use the DoTA dataset to fine-tune the whole model. The training objective is as follows:

$$\mathcal{L} = - \sum_{t=1}^n \log p(y_t | y_{<t}, \mathbf{I}; \theta) \quad (9)$$

where θ denotes the parameters of the entire model.

4 DoTA Dataset

To facilitate the research community, we build a novel large-scale Document image machine Translation dataset of ArXiv articles in markdown format (**DoTA**). Specifically, we randomly select 18,496 papers published on arXiv from 2020 to 2023 and download the corresponding PDF files and L^AT_EX source codes. The category distribution of these articles is shown in Table 2.

Category	# Articles	Percentage (%)
Physics	6,754	36.5
Mathematics	3,035	16.4
Computer Science	6,536	35.3
Quantitative Biology	225	1.2
Quantitative Finance	88	0.5
Statistics	563	3.0
Electrical Engineering	1,193	6.5
Economics	102	0.6
Total	18,496	100

Table 2: Category distribution of DoTA dataset. # Articles denotes the number of articles.

Following Blecher et al. (2023), we employ *LaTeXML* to process source codes, transforming them into HTML files, and then to markdown files. Throughout this process, formulas and tables are retained in their original L^AT_EX source code, while figures are omitted. Then, we split the markdown files according to the page breaks in the PDF file and rasterize each page as an image to create the final paired dataset. Several techniques are used to ensure the quality of the dataset. See the Appendix A for details.

For translation, we translate English texts in the markdown files of the train set into Chinese with Google Translate API. The valid set and test set are translated by professional translators. For formulas and tables, we utilize special tokens to substitute them before the translation process, reinstating them after the translation is completed.

To guarantee translation quality, we use COMET (Rei et al., 2020) to score each translation pair and remove the lowest scoring 10% of pairs.⁶ Besides, we randomly sample 1000 translation pairs and employ three professional translators to evaluate these translation texts on a scale of 1 to 5, with 1 indicating the lowest quality and 5 indicating the highest, in terms of fluency and fidelity. The average score of fluency and fidelity are 3.70 and 3.53,

⁶We use `wmt22-cometkiwi-da` in reference-free mode.

	Train Set	Valid Set	Test Set
# Articles	18,229	147	120
# Page Images	124,338	1,004	1,003
# AvgTokens (En)	466	474	447
# AvgTokens (Zh)	1,307	1,342	1,258

Table 3: The statistics of the DoTA dataset. Each image in the dataset is paired with an English markdown file and a Chinese markdown file. # Articles and # Page Images denote the number of articles and the number of page images, respectively. # AvgTokens (En) and # AvgTokens (Zh) denote the average number of English words and the average number of Chinese characters in each image, separately.

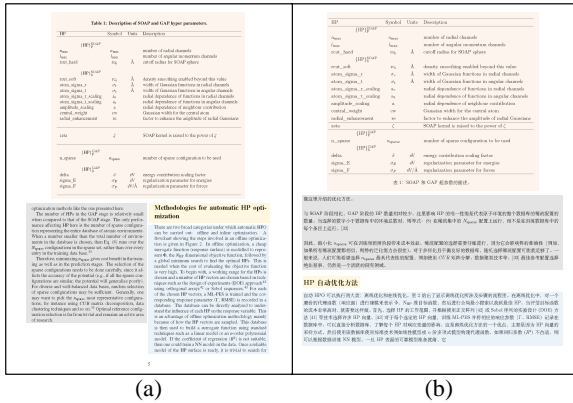


Figure 3: Dataset sample. (a) is the original document image. (b) is the translated text in markdown format after rendering. The blocks with the same color represent the corresponding original text and translated text.

respectively, indicating an acceptable translation quality.

Finally, 126,345 paired images, English markdown files, and their corresponding Chinese translations are obtained. More details are illustrated in Table 3. Figure 3 presents a sample of our dataset.

5 Experiments

5.1 Dataset & Metrics

We use the DoTA dataset to train and evaluate our method and baselines. In addition, we select samples of physics and computer science domains from the test set to form domain-specific test sets.

The evaluation metrics are as follows:

BLEU We calculate the BLEU score (Papineni et al., 2002) between raw prediction and reference, which contains plain texts, formulas, and tables in markdown format.

BLEU-PT As the translated texts contain formulas and tables in markdown format that influence the BLEU computing, we remove them from both

reference texts and predictions, keep the plain texts, and calculate BLEU noted as BLEU-PT.

STEDS As the markdown format contains the hierarchical structure of document images, relying solely on a string-based comparison metric, like BLEU, can be insufficient. We adopt structure tree-edit-distance-based similarity (STEDS) (Zhong et al., 2020) as an additional evaluation metric to measure the performance of produced layout results. We apply a rule-based method to convert each of the prediction and reference texts into a tree T . Each inner node is a title and each leaf node is a paragraph, a line formula, or a table. Then, the tree edit distance (TED) is calculated through the ZSS algorithm (Zhang and Shasha, 1989).⁷ The STEDS can be formulated as:

$$\text{STEDS}(T_{\text{pred}}, T_{\text{ref}}) = 1 - \frac{\text{TED}(T_{\text{pred}}, T_{\text{ref}})}{\max(|T_{\text{pred}}|, |T_{\text{ref}}|)} \quad (10)$$

where $|T|$ denotes the number of nodes in T .

5.2 Settings

Pre-trained Models Selecting The encoder of the pre-trained model Nougat (Blecher et al., 2023) is used to initialize the OCR encoder in DIMTDA. The layout encoder is initialized from a pre-trained layout analysis model DiT (Li et al., 2022b). We follow the vanilla Transformer-base (Vaswani et al., 2017) setting, pre-train an English-Chinese translation model on UN Corpus En-Zh (Ziemski et al., 2016), and use the pre-trained decoder to initialize the translation decoder in DIMTDA.

Other Settings Each Qformer is a one-layer Transformer with 1024 queries and is randomly initialized. During fine-tuning, we employ a linear decay learning rate schedule with a learning rate of $5e-5$. The maximum number of training epochs is 30 and batch size is 128. For inference, we use beam search with a beam size of 4. More detailed experimental settings are in Appendix B.

5.3 Baselines

We compare our method against both cascade methods and end-to-end text image translation methods.

Text-only MT We use the DoTA dataset to fine-tune the Transformer-base model pre-trained on UN Corpus En-Zh.

Docxchain + Tesseract + Trans (DTT) This cascade system first employs a layout analysis

⁷We only consider the differences in tree structure and do not take into account the similarity of content for nodes at the same position.

		All Test data			Physics Domain			Computer Science Domain			# Param (M) ↓	Time (s/page) ↓
		BLEU	BLEU-PT	STEDS	BLEU	BLEU-PT	STEDS	BLEU	BLEU-PT	STEDS		
1	Text-only MT	47.61	54.16	92.89	47.32	52.85	95.09	51.93	55.76	93.68	99.5	8.81
Cascade Baselines												
2	DTT	35.58	41.75	75.83	31.28	39.09	74.23	38.52	42.51	73.73	99.5 + α	12.46
3	NT	43.37	50.79	88.16	44.16	49.73	90.19	48.57	52.88	88.78	346.9	17.03
End-to-end TIMT Baselines (Document-level)												
4	ItNet	3.84	2.27	48.46	3.68	3.60	47.66	4.10	4.35	48.69	97.5	8.43
5	E2ETIT	1.51	1.69	32.90	1.29	0.81	32.66	1.79	0.29	34.84	122.0	8.19
6	PEIT	5.81	4.52	55.79	5.06	5.72	51.65	6.62	6.71	60.73	135.1	2.57
End-to-end TIMT Baselines (Text-line-level)												
7	ItNet	21.75	23.52	75.83	19.39	21.19	74.23	26.22	28.36	73.73	97.5 + β	7.20
8	E2ETIT	17.42	17.74	75.83	15.91	15.44	74.23	20.82	21.95	73.73	122.0 + β	7.59
9	PEIT	27.43	31.29	75.83	24.80	29.10	74.23	32.75	36.71	73.73	135.1 + β	2.42
Pre-trained Model Assembling												
10	Base	37.60	40.85	83.08	36.55	38.22	83.73	39.97	41.52	81.03	127.6	9.16
11	Base (Random)	1.26	1.69	34.31	1.61	0.28	33.67	1.45	0.36	31.09	127.6	12.95
12	Addition	38.26	39.66	83.67	37.55	37.59	83.86	39.30	40.88	81.36	242.6	10.15
13	Attention	31.12	32.30	79.31	30.55	30.96	79.54	31.23	31.44	74.69	245.7	11.12
14	Concatenation	37.99	38.97	83.73	37.69	39.77	83.88	39.53	40.03	82.33	242.6	10.76
15	DIMTDA	38.68	42.34	84.44	36.75	40.38	85.06	39.33	42.06	81.33	242.6	9.82

Table 4: Results on the English-Chinese test set. *Time* is the average inference time on a single V100 GPU. α denotes the parameters of the layout analysis model and OCR model. β denotes the parameters of the parameters of the layout analysis model and sentence splitting model. *Random* means random initialization.

model (Yao, 2023) to recognize each layout block and its label. Then, the OCR tool tesseract⁸ extracts texts from each layout block. Finally, the text-only MT mentioned above is used to do translation.

Nougat + Trans (NT) We utilize the Nougat model (Blecher et al., 2023) for combined layout analysis and OCR, which outputs the recognized text in markdown format. Additionally, the text-only MT is employed for translation.

For end-to-end TIMT baselines, we first make a **Document-level** experiment: directly inputting the entire image into the model for translation. Furthermore, we notice that all end-to-end TIMT baselines are designed for text-line-level images, rather than document-level ones. To make a fair comparison, we also conduct a **Text-line-level** experiment: utilizing the pre-trained layout analysis model and text-line detection model to extract text-line images, followed by text-line-level image machine translation and reconstruction into the original document.

ItNet (Jain et al., 2021) This is an end-to-end TIMT system. It first pre-trains a vanilla Transformer on a text parallel dataset. Then, the combination of the image encoder and pre-trained decoder is fine-tuned.

E2ETIT (Ma et al., 2022) This end-to-end model uses a TPSNet and a ResNet as an image encoder combined with a Transformer decoder and utilizes text translation as an auxiliary task.

PEIT (Zhu et al., 2023) This end-to-end TIT system employs a vision-text representation aligner and a cross-model regularize to bridge the modality gap between visual inputs and textual inputs.

Besides, we also implement four conventional model assembling methods for comparison.

Base The feature sequences of the OCR encoder are directly sent to the translation decoder for cross-attention computation. Random initialization is made for comparison.

Addition The outputs of the two Qformers are directly combined through element-wise addition, which can be formulated as:

$$\mathbf{H}_{\text{fuse}} = \text{FFN}(0.5 \odot \mathbf{H}_{\text{ocr}} + 0.5 \odot \mathbf{H}_{\text{lay}}) \quad (11)$$

Attention We employ eight attention heads for attention computation, configuring the OCR Qformer’s output for Q, and utilizing the layout Qformer’s output for K/V.

Concatenation The output sequence of the layout Qformer is concatenated after that of the OCR Qformer. This can be formulated as:

$$\mathbf{H}_{\text{fuse}} = \text{FFN}([\mathbf{H}_{\text{ocr}} : \mathbf{H}_{\text{lay}}]) \quad (12)$$

6 Results & Analysis

6.1 Main Results

Table 4 reports the performance of all models. We can observe (line 2 vs. line 15) that our method outperforms the cascade method DTT, while the inference time of DIMTDA is only 79% that of the

⁸<https://github.com/tesseract-ocr/tesseract/>

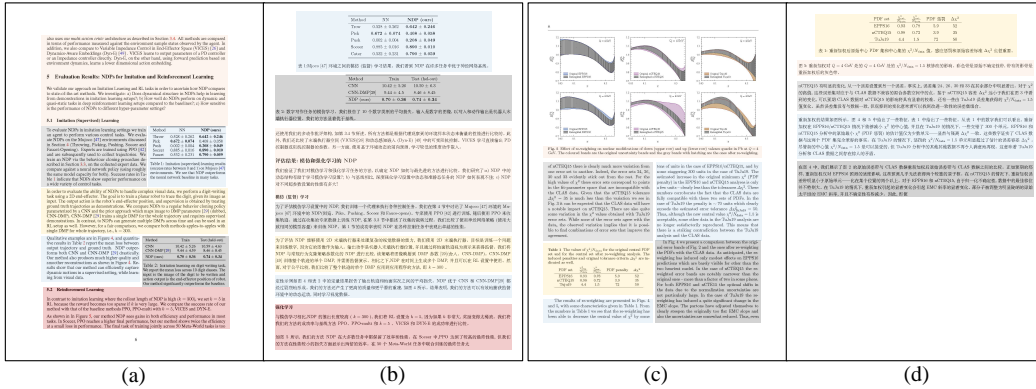


Figure 4: The output samples of DIMTDA. (a) and (c) are the original document images. (b) and (d) are the output translated texts in markdown format after rendering.

		All Valid Data			Simple Layout Set			Complex Layout Set		
		BLEU	BLEU-PT	STEDS	BLEU	BLEU-PT	STEDS	BLEU	BLEU-PT	STEDS
1	DIMTDA	38.71	42.58	84.52	55.24	55.26	90.54	30.30	35.16	84.57
2	w/o Layout Encoder	37.67	39.88	83.55	54.98	55.65	90.75	26.40	30.96	79.83
3	w/o OCR Encoder	2.39	1.11	49.46	1.26	1.23	53.02	3.71	0.80	38.82
4	w/o Model Assembler	36.97	41.08	83.73	53.41	53.89	90.26	29.77	33.11	82.80
5	Base	37.25	40.58	83.27	54.68	54.88	90.76	27.44	33.00	81.21

Table 5: Ablation study results of our model on the English-Chinese valid set.

DTT. Although DIMTDA is inferior to the cascade method NT in translation performance (line 3 vs. line 15), the parameter of DIMTDA is reduced by 30% compared to NT, and inference time decreases by 42%.

Besides, our method outperforms all the end-to-end TIMT baselines on both document-level and text-line-level settings. Although all TIMT baselines show a significant improvement in performance under the text-line-level setting compared to the document-level setting, DIMTDA still surpasses the highest-performing TIMT model (line 9 vs. line 15) by a margin of 11.25 BLEU and 8.61 STEDS points on all test data.

The comparison between line 10 and line 15 reveals an increase of 1.08 BLEU and 1.36 STEDS points on all test data, underscoring the significance of layout information. To validate the effectiveness of the proposed dynamic fusion mechanism in this paper, we conduct comparative experiments with addition, attention, and concatenation fusion methods. Results in Table 4 lines 12-15 indicate that the dynamic fusion mechanism surpasses other model fusion methods.

The output samples of DIMTDA are shown in Figure 4. More samples are in Appendix C.

6.2 Ablation Study

To investigate the effectiveness of different modules, we conduct ablation experiments. Besides, to examine whether the model effectively leverages layout information, we select two subsets from the valid set, one with simple layouts and the other with complex layouts.⁹ The results are in Table 5.

w/o Layout Encoder We remove the layout encoder, layout Qformer, and gate module. By comparing line 1 and 2, the performance declines on both the simple and complex layout test sets, with a more pronounced decrease on the complex layout test set. It suggests that the layout information indeed enhances the model’s understanding of layout structure, with more notable improvements observed on images with complex layouts.

w/o OCR Encoder We remove the OCR encoder, OCR Qformer, and gate module. As the result of line 3 suggests, without textual information, only the layout information can hardly guide the decoder to generate translation texts.

w/o Model Assembler We remove the model assembler and concatenate the output of the layout encoder after that of the OCR encoder, which is

⁹We transform samples from the valid set into trees, selecting the 100 trees with the fewest nodes as simple layout set and the 100 trees with the most nodes as complex layout set.

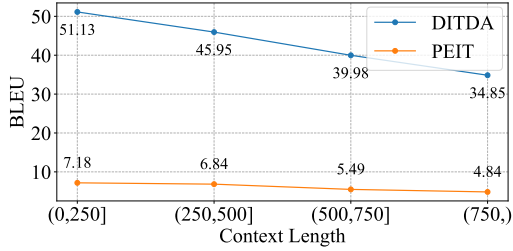


Figure 5: Results of different context lengths.

formulated as:

$$\mathbf{H}_{\text{fuse}} = \text{FFN}([\mathbf{H}_{\text{ocr}} : \text{FFN}(\mathbf{H}_{\text{lay}})]) \quad (13)$$

The comparison between line 1 and line 4 shows a decline of 1.74 BLEU and 0.79 STEDS points on all valid data, which demonstrates the effectiveness of the model assembler.

6.3 Effect of Context Length

To investigate the impact of context length in images on the model, we select samples from the valid set within different lengths.¹⁰ Results are shown in Figure 5, where the X-axis and Y-axis represent context length and BLEU score, respectively.

As the context length increases, the performance of DIMTDA gradually declines but remains significantly superior to PEIT, which demonstrates DIMTDA’s capability to model document images with long context.

6.4 Effect of Different Initialization

There are various OCR pre-trained models, layout pre-trained models, and language models. Therefore, to investigate the effect of different module structures and parameter initializations on the model, we substitute the OCR encoder, layout encoder, and translation decoder with TrOCR (Li et al., 2023b), BEiT (Bao et al., 2022), and mBART (Liu et al., 2020), respectively. We also retain the structure of the model from the main experiment but perform separate random initializations for each module. Table 6 presents the results of different settings on the DoTA valid set.

By comparing lines 1-3, it can be observed that the OCR encoder significantly impacts the model’s performance, and nougat pre-trained on document

¹⁰To shield the impact of layout difference and keep the layout consistent across different lengths, we select images with a single column and without line formulas, tables, or figures. Context length refers to the number of English words in the image.

		BLEU	STEDS
1	Nougat + DiT + Trans-dec	38.71	84.52
2	TrOCR + DiT + Trans-dec	2.45	51.20
3	Nougat (Random) + DiT + Trans-dec	2.11	46.99
4	Nougat + BEiT + Trans-dec	32.89	78.59
5	Nougat + DiT (Random) + Trans-dec	35.09	81.47
6	Nougat + DiT + mBART-dec	34.11	80.88
7	Nougat + DiT + Trans-dec (Random)	6.25	59.31

Table 6: Results of different initialization settings. *Trans-dec* and *mBART-dec* denote the decoder of pre-trained Transformer-base and mBART respectively. *Random* means random initialization.

images achieves the best performance. Regarding the layout encoder, the results (line 1, 4, and 5) indicate that DiT initialization is more suitable for the DIMT2Markdown task. As for the decoder, the use of language model decoder initialization (line 6) yields inferior results compared to the initialization with a translation model (line 1). Therefore, our experimental results suggest that it is better to initialize the OCR encoder parameters with Nougat encoder, layout encoder with DiT, and translation decoder with pre-trained translation model decoder.

6.5 Effect of Hyper Parameter

To explore the impact of different numbers of queries, we vary the number of queries from 16 to 2048. The results on the valid set are illustrated in Figure 6, where the X-axis and Y-axis represent the number of queries and scores, separately.

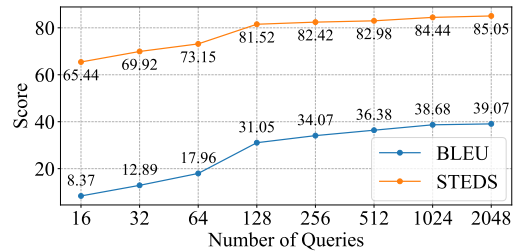


Figure 6: Results of the different number of queries.

It can be observed that when the number of queries is less than 128, there is a significant improvement in the model’s performance with an increase in query quantity. However, after reaching 1024 queries, the improvement becomes marginal. To balance the number of parameters and performance, we set 1024 queries in the main experiment.

		Test Data			Physics Domain			Computer Science Domain		
		BLEU	BLEU-PT	STEDS	BLEU	BLEU-PT	STEDS	BLEU	BLEU-PT	STEDS
1	Text-only MT	48.59	57.41	91.10	44.62	51.97	98.98	57.05	58.28	92.06
2	GPT-4V	21.51	26.89	57.77	20.58	22.77	62.24	17.95	22.51	63.33
3	Gemini	26.88	29.34	54.00	27.09	30.69	58.89	32.30	37.32	61.63
4	DIMTDA	47.39	49.26	88.21	25.96	28.14	80.33	45.43	47.19	88.39

Table 7: Results on comparison with commercial MLLMs.

6.6 Comparison with Multimodal Large Language Models

With the rapid development of multimodal large language models (MLLM), some commercial MLLMs (Yang et al., 2023; Team et al., 2023) have also demonstrated the capability of understanding text-rich document images. To assess their ability to accomplish the DIMT2Markdown task, we randomly choose 20 samples from each test set in the main experiment, then prompt GPT-4V (Yang et al., 2023) and Gemini (Team et al., 2023) with "Output the Chinese translations of this image in markdown format." As the output format of MLLMs may be unstable, we filter the English parts of the output and only keep the Chinese parts.

Table 7 reveals that both GPT-4V and Gemini exhibit inferior performance compared to DITDA. The commercial MLLMs are not trained on the DoTA dataset, so their output format is different from the reference. Besides, despite their generations being fluent and faithful, their actual performance can not be fully reflected by BLEU and STEDS scores.

6.7 Evaluation on Other Languages

We evaluate our DIMTDA on English-French and English-German DIMT2Markdown tasks. The MT models are pre-trained on UN Corpus En-Fr and WMT14 En-De. We employ the Google Translate API to obtain English-French and English-German translation pairs for the DoTA dataset. The rest of the settings remain the same as the main experiment. Table 8 demonstrates the effectiveness of DIMTDA on other languages.

		En-Fr		En-De	
		BLEU	STEDS	BLEU	STEDS
1	Text-only MT	59.68	95.93	49.25	96.04
2	DTT	42.79	75.59	32.65	75.59
3	NT	55.82	90.77	43.73	89.92
4	DIMTDA	45.82	84.84	37.83	85.92

Table 8: Results on the English-French and English-German test sets.

7 Conclusion

In this paper, we propose a novel text image machine translation task, DIMT2Markdown, taking a source document image as input and outputting translations in markdown format, which broadens the research of text image machine translation. We also construct a large-scale dataset named DoTA for this task. Besides, a novel end-to-end framework, DIMTDA, is introduced, which dynamically assembles multi-pre-trained models to fulfill this task. Comprehensive experiments demonstrate promising prospects for the direct translation of document images into markdown-formatted texts and the effectiveness of DIMTDA.

Limitations

Although DIMTDA achieves end-to-end image machine translation on text-rich document images, the performance on images with line formulas and tables is unsatisfactory, which may be caused by dense numerical information, abbreviated alphanumeric characters, and complex table structure. We will consider modeling tables and formulas specifically in the future.

Acknowledgements

First, We thank anonymous reviewers for helpful suggestions. Second, we thank all the annotators and volunteers for constructing the dataset and making the human evaluation. This work is supported by the National Natural Science Foundation of China (No. 62106265).

References

- Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. 2022. *BEit: BERT pre-training of image transformers*. In *International Conference on Learning Representations*.
- Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. 2023. Nougat: Neural optical understanding for academic documents. *arXiv preprint arXiv:2308.13418*.

- Ozan Caglayan, Menekse Kuyu, Mustafa Sercan Amac, Pranava Madhyastha, Erkut Erdem, Aykut Erdem, and Lucia Specia. 2021. [Cross-lingual visual pre-training for multimodal machine translation](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1317–1324, Online. Association for Computational Linguistics.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.
- Wenyu Guo, Qingkai Fang, Dong Yu, and Yang Feng. 2023. [Bridging the gap between synthetic and authentic images for multimodal machine translation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2863–2874, Singapore. Association for Computational Linguistics.
- Xin Huang, Jiajun Zhang, and Chengqing Zong. 2021. [Entity-level cross-modal learning improves multimodal machine translation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1067–1080, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Puneet Jain, Orhan Firat, Qi Ge, and Sihang Liang. 2021. Image translation network.
- Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoon Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. In *European Conference on Computer Vision*, pages 498–517. Springer.
- Zhibin Lan, Jiawei Yu, Xiang Li, Wen Zhang, Jian Luan, Bin Wang, Degen Huang, and Jinsong Su. 2023. [Exploring better text image translation with multimodal codebook](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3479–3491, Toronto, Canada. Association for Computational Linguistics.
- Bei Li, Chuanhao Lv, Zefan Zhou, Tao Zhou, Tong Xiao, Anxiang Ma, and JingBo Zhu. 2022a. [On vision features in multimodal machine translation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6327–6337, Dublin, Ireland. Association for Computational Linguistics.
- Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. 2022b. Dit: Self-supervised pre-training for document image transformer. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3530–3539.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. [Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models](#). *arXiv preprint arXiv:2301.12597*.
- Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. 2023b. Trocr: Transformer-based optical character recognition with pre-trained models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13094–13102.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. [Multilingual denoising pre-training for neural machine translation](#). *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022.
- Tengchao Lv, Yupan Huang, Jingye Chen, Lei Cui, Shuming Ma, Yaoyao Chang, Shaohan Huang, Wenhui Wang, Li Dong, Weiyao Luo, et al. 2023. [Kosmos-2.5: A multimodal literate model](#). *arXiv preprint arXiv:2309.11419*.
- Cong Ma, Xu Han, Linghui Wu, Yaping Zhang, Yang Zhao, Yu Zhou, and Chengqing Zong. 2023a. [Modal contrastive learning based end-to-end text image machine translation](#). *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.
- Cong Ma, Yaping Zhang, Mei Tu, Xu Han, Linghui Wu, Yang Zhao, and Yu Zhou. 2022. [Improving end-to-end text image translation from the auxiliary text translation task](#). In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 1664–1670. IEEE.
- Cong Ma, Yaping Zhang, Mei Tu, Yang Zhao, Yu Zhou, and Chengqing Zong. 2023b. [E2timt: Efficient and effective modal adapter for text image machine translation](#). In *The 17th International Conference on Document Analysis and Recognition (ICDAR)*, pages 70–88.
- Cong Ma, Yaping Zhang, Mei Tu, Yang Zhao, Yu Zhou, and Chengqing Zong. 2023c. [Multi-teacher knowledge distillation for text image machine translation](#). In *The 17th International Conference on Document Analysis and Recognition (ICDAR)*, pages 484–501.
- Elman Mansimov, Mitchell Stern, Mia Chen, Orhan Firat, Jakob Uszkoreit, and Puneet Jain. 2020. [Towards end-to-end in-image neural machine translation](#). In *Proceedings of the First International Workshop on Natural Language Processing Beyond Text*, pages 70–74, Online. Association for Computational Linguistics.

- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. [COMET: A neural framework for MT evaluation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.
- Nilesh P Sable, Priya Shelke, Ninad Deogaonkar, Nachiket Joshi, Rudra Kabadi, and Tushar Joshi. 2023. Doc-handler: Document scanner, manipulator, and translator based on image and natural language processing. In *2023 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pages 1–6. IEEE.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. The dawn of Imms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1.
- Cong Yao. 2023. Docxchain: A powerful open-source toolchain for document parsing and beyond. *arXiv preprint arXiv:2310.12430*.
- Shaowei Yao and Xiaojun Wan. 2020. [Multimodal transformer for multimodal machine translation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4346–4350, Online. Association for Computational Linguistics.
- Kaizhong Zhang and Dennis Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6):1245–1262.
- Zhiyang Zhang, Yaping Zhang, Yupu Liang, Lu Xiang, Yang Zhao, Yu Zhou, and Chengqing Zong. 2023. [LayoutDIT: Layout-aware end-to-end document image translation with multi-step conductive decoder](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10043–10053, Singapore. Association for Computational Linguistics.
- Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. 2020. Image-based table recognition: data, model, and evaluation. In *European conference on computer vision*, pages 564–580. Springer.
- Shaolin Zhu, Shangjie Li, Yikun Lei, and Deyi Xiong. 2023. [PEIT: Bridging the modality gap with pre-trained models for end-to-end image translation](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13433–13447, Toronto, Canada. Association for Computational Linguistics.
- Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. [The United Nations parallel corpus v1.0](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA).

Appendix

A Dataset Quality Control

As the collected articles are mainly scientific papers, there are about 10% of images that only contain references, which may introduce noise into the dataset. We remove these image-markdown pairs from the dataset. Besides, we divide the dataset into a train set, a valid set, and a test set, and ensure that images of the same article are not divided into different sets to avoid data leakage.

B Setting Details

We segment the Chinese texts with jieba and apply WordPiece to segment both English and Chinese texts and the vocabulary size of both English and Chinese is 52K. We use the pre-trained OCR model Nougat’s encoder (Blecher et al., 2023) to initialize the OCR encoder. It is a Swin Transformer-based (Liu et al., 2021) encoder and the layer numbers and window size are {2, 2, 14, 2} and 7. The hidden size of each layer is 1024 and the patch size is 4. The input image size for the OCR encoder is 896×672. The layout encoder is initialized by a pre-trained DiT model (Li et al., 2022b) which has 12 transformer layers. Each layer has 12 attention heads and the hidden size of each layer is 768. The input image size for the layout encoder is 224×224. As for Qformer, each Qformer is a one-layer Transformer with 1024 query tokens. Its hidden size and number of attention heads are 1024 and 8. We follow the vanilla Transformer-base (Vaswani et al., 2017) setting and pre-train an English-Chinese translation model on UN Corpus. We set the decoder’s max length and max position embeddings to 1536 to cover most input texts.

The figure shows a 4-stage pipeline for image and text processing. Stage 1: Input (Image and Text). Stage 2: Pre-processing (Image and Text). Stage 3: Feature Extraction (Image and Text). Stage 4: Output (Image and Text).

Figure 7: Output samples of DIMTDA. The figure shows two columns of image pairs. The left column shows the input document image, and the right column shows the output translation in markdown format after rendering.

Figure 8: Evaluation results of DIMTDA. The figure shows a table with columns for 'Metric', 'Value', and 'Unit'. The table contains data for various metrics such as Accuracy, Precision, Recall, and F1-score.

Metric	Value	Unit
Accuracy	0.95	%
Precision	0.92	%
Recall	0.98	%
F1-score	0.94	%

Figure 9: Comparison of DIMTDA with other methods. The figure shows a table with columns for 'Method', 'Accuracy', 'Precision', 'Recall', and 'F1-score'. The table compares DIMTDA with other methods such as BERT, GPT-3, and others.

Method	Accuracy	Precision	Recall	F1-score
BERT	0.90	0.88	0.92	0.90
GPT-3	0.92	0.90	0.94	0.92
DIMTDA	0.95	0.92	0.98	0.94

Figure 10: Performance of DIMTDA on different tasks. The figure shows a table with columns for 'Task', 'Accuracy', 'Precision', 'Recall', and 'F1-score'. The table shows the performance of DIMTDA on various tasks such as Text Classification, Image Classification, and Image Captioning.

Task	Accuracy	Precision	Recall	F1-score
Text Classification	0.92	0.90	0.94	0.92
Image Classification	0.94	0.92	0.96	0.94
Image Captioning	0.91	0.89	0.93	0.91

Figure 11: Ablation study of DIMTDA. The figure shows a table with columns for 'Component', 'Accuracy', 'Precision', 'Recall', and 'F1-score'. The table shows the impact of different components on the performance of DIMTDA.

Component	Accuracy	Precision	Recall	F1-score
Full Model	0.95	0.92	0.98	0.94
Without Pre-processing	0.90	0.88	0.92	0.90
Without Feature Extraction	0.88	0.86	0.90	0.88

Figure 12: Comparison of DIMTDA with other methods on different tasks. The figure shows a table with columns for 'Method', 'Task', 'Accuracy', 'Precision', 'Recall', and 'F1-score'. The table compares DIMTDA with other methods on various tasks.

Method	Task	Accuracy	Precision	Recall	F1-score
DIMTDA	Text Classification	0.92	0.90	0.94	0.92
BERT	Text Classification	0.90	0.88	0.92	0.90
DIMTDA	Image Classification	0.94	0.92	0.96	0.94

Figure 13: Ablation study of DIMTDA on different tasks. The figure shows a table with columns for 'Component', 'Task', 'Accuracy', 'Precision', 'Recall', and 'F1-score'. The table shows the impact of different components on the performance of DIMTDA on various tasks.

Component	Task	Accuracy	Precision	Recall	F1-score
Full Model	Text Classification	0.92	0.90	0.94	0.92
Without Pre-processing	Text Classification	0.90	0.88	0.92	0.90
Without Feature Extraction	Text Classification	0.88	0.86	0.90	0.88

Figure 14: Performance of DIMTDA on different tasks. The figure shows a table with columns for 'Task', 'Accuracy', 'Precision', 'Recall', and 'F1-score'. The table shows the performance of DIMTDA on various tasks.

Task	Accuracy	Precision	Recall	F1-score
Text Classification	0.92	0.90	0.94	0.92
Image Classification	0.94	0.92	0.96	0.94
Image Captioning	0.91	0.89	0.93	0.91

Figure 15: Ablation study of DIMTDA on different tasks. The figure shows a table with columns for 'Component', 'Task', 'Accuracy', 'Precision', 'Recall', and 'F1-score'. The table shows the impact of different components on the performance of DIMTDA on various tasks.

Component	Task	Accuracy	Precision	Recall	F1-score
Full Model	Text Classification	0.92	0.90	0.94	0.92
Without Pre-processing	Text Classification	0.90	0.88	0.92	0.90
Without Feature Extraction	Text Classification	0.88	0.86	0.90	0.88

Figure 16: Comparison of DIMTDA with other methods on different tasks. The figure shows a table with columns for 'Method', 'Task', 'Accuracy', 'Precision', 'Recall', and 'F1-score'. The table compares DIMTDA with other methods on various tasks.

Method	Task	Accuracy	Precision	Recall	F1-score
DIMTDA	Text Classification	0.92	0.90	0.94	0.92
BERT	Text Classification	0.90	0.88	0.92	0.90
DIMTDA	Image Classification	0.94	0.92	0.96	0.94

Figure 17: Ablation study of DIMTDA on different tasks. The figure shows a table with columns for 'Component', 'Task', 'Accuracy', 'Precision', 'Recall', and 'F1-score'. The table shows the impact of different components on the performance of DIMTDA on various tasks.

Component	Task	Accuracy	Precision	Recall	F1-score
Full Model	Text Classification	0.92	0.90	0.94	0.92
Without Pre-processing	Text Classification	0.90	0.88	0.92	0.90
Without Feature Extraction	Text Classification	0.88	0.86	0.90	0.88

Figure 7: The output samples of DIMTDA. For each image pair, the left one is the input document image, and the right one is the output translations in markdown format after rendering.

During translation model pre-training, the maximum training step is 100K and the maximum token per batch is 4096. A linear decay learning rate schedule with a learning rate of 7e-4 and a warmup ratio of 0.05 is used. During fine-tuning, the maximum number of training epochs is 30 and the batch size is 128. We use a linear decay learning rate schedule with a learning rate of 5e-5 and a warmup ratio of 0.05. We early-stop the fine-tuning if the loss on the valid set does not decrease for 1 epoch. We use Adam optimizer with $\beta_1=0.9$, $\beta_2=0.999$, $\epsilon=1e-8$ for both pre-training and fine-tuning. For inference, we use beam search with 4 beams.

C Output Samples

More samples are shown in Figure 7.