

# Translatotron-V(ision): An End-to-End Model for In-Image Machine Translation

Zhibin Lan<sup>1,3\*</sup>, Liqiang Niu<sup>2</sup>, Fandong Meng<sup>2</sup>, Jie Zhou<sup>2</sup>, Min Zhang<sup>4</sup>, Jinsong Su<sup>1,3†</sup>

<sup>1</sup>School of Informatics, Xiamen University, China

<sup>2</sup>Pattern Recognition Center, WeChat AI, Tencent Inc, China

<sup>3</sup>Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan (Xiamen University), Ministry of Culture and Tourism, China

<sup>4</sup>Institute of Computer Science and Technology, Soochow University, China

lanzhibin@stu.xmu.edu.cn jssu@xmu.edu.cn

## Abstract

In-image machine translation (IIMT) aims to translate an image containing texts in source language into an image containing translations in target language. In this regard, conventional cascaded methods suffer from issues such as error propagation, massive parameters, and difficulties in deployment and retaining visual characteristics of the input image. Thus, constructing end-to-end models has become an option, which, however, faces two main challenges: 1) the huge modeling burden, as it is required to simultaneously learn alignment across languages and preserve the visual characteristics of the input image; 2) the difficulties of directly predicting excessively lengthy pixel sequences. In this paper, we propose *Translatotron-V(ision)*, an end-to-end IIMT model consisting of four modules. In addition to an image encoder, and an image decoder, our model contains a target text decoder and an image tokenizer. Among them, the target text decoder is used to alleviate the language alignment burden, and the image tokenizer converts long sequences of pixels into shorter sequences of visual tokens, preventing the model from focusing on low-level visual features. Besides, we present a two-stage training framework for our model to assist the model in learning alignment across modalities and languages. Finally, we propose a location-aware evaluation metric called Structure-BLEU to assess the translation quality of the generated images. Experimental results demonstrate that our model achieves competitive performance compared to cascaded models with only 70.9% of parameters, and significantly outperforms the pixel-level end-to-end IIMT model.<sup>1</sup>

## 1 Introduction

In recent years, significant advancements have been achieved in natural language processing (NLP) and

\*Work was done when Zhibin Lan was interning at Pattern Recognition Center, WeChat AI, Tencent Inc, China.

†Corresponding author.

<sup>1</sup>Our code and dataset can be found at <https://github.com/DeepLearnXMU/translatotron-v>.

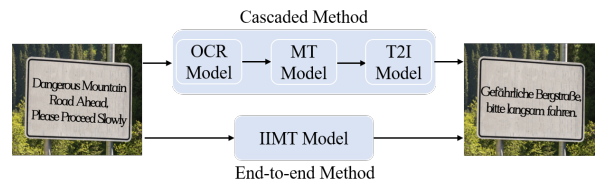


Figure 1: The illustration of two paradigms of IIMT.

computer vision (CV), largely due to the evolution of deep learning. As a combined direction of these two fields, in-image machine translation (IIMT) aims to convert an image containing texts in source language into another image containing the translations in target language, which has significant research value and practical applications. It not only helps us understand the fusion mechanism of multimodal and multilingual information, but also finds widespread applications in daily life. For instance, IIMT can effortlessly enable foreign travelers to read signs written in other languages.

As shown in Figure 1, current IIMT systems are divided into two paradigms: cascaded and end-to-end. The first one relies on cascading multiple models, including an optical character recognition (OCR) model, a machine translation (MT) model, and a text-to-image (T2I) model. However, this paradigm suffers from error propagation, massive parameters, and difficulties in deployment and retaining visual characteristics of the input image. By contrast, end-to-end methods (Mansimov et al., 2020; Tian et al., 2023) integrate different models into one IIMT model and conduct end-to-end training. Thus, they have potential advantages over cascaded systems in aspects of avoiding error propagation, reduced parameters, and ease of deployment. Particularly, they are naturally capable of retaining visual characteristics from the input image during translation, e.g. maintaining background, text location, font, etc.

Despite the above advantages, the end-to-end IIMT models still face two major challenges: 1) the huge modeling burden, since they are required

to not only learn alignment between two languages but also the visual characteristics of the input image; 2) the difficulties of directly predicting excessively lengthy pixel sequences, which are low-level and involve a large search space (Ramesh et al., 2021; Yu et al., 2022b).

To the best of our knowledge, (Mansimov et al., 2020) and (Tian et al., 2023) are the only two attempts to explore end-to-end IIMT. However, the former is directly based on pixel prediction, resulting in significantly lower translation quality compared to the cascaded models, while the latter requires converting RGB images to grayscale ones, losing visual characteristics. Besides, both of them can only handle images containing single-line text. These defects make them still far from real-world applications.

In this paper, we propose *Translatotron-V(ision)*, the first end-to-end IIMT model capable of generating RGB images, achieving comparable performance to cascaded models with only 70.9% of parameters. As shown in Figure 2, our model consists of four modules: 1) an *image encoder* that represents the semantics of the image as a sequence of visual vectors; 2) a *target text decoder* that utilizes the visual vector sequence to predict the text translation, which can effectively reduce the modeling burden on the image decoder; 3) an *image decoder* that generates the visual tokens of the target image based on the visual and linguistic information generated from the image encoder and target text decoder, respectively; 4) an *image tokenizer* that converts the image into discrete visual tokens and can reconstruct the image from these visual tokens. By converting the image into visual tokens, the image decoder only needs to predict visual tokens, rather than excessively lengthy pixel sequences, which allows the model to avoid spending too much capacity capturing low-level visual features.

Furthermore, as illustrated in Figure 3, we propose a training framework for our model, consisting of two stages. First, we utilize large-scale unlabeled images to train the image tokenizer through an image reconstruction task. Then, we freeze the image tokenizer and train other modules using IIMT dataset. Inspired by end-to-end speech translation (Jia et al., 2019), we introduce multi-task learning at this stage. The auxiliary tasks include OCR and text image translation (TIT), assisting the model in learning alignment across different modalities and languages. Particularly, we intro-

duce a knowledge distillation method to reduce the difficulty of the end-to-end model directly learning from ground-truth labels.

Due to the absence of publicly available IIMT datasets, we use IWSLT14 German-English (Cettolo et al., 2014) to synthesize a dataset for this task. Note that unlike previous works (Mansimov et al., 2020; Tian et al., 2023) only focus on images containing single-line text, the images in our dataset are more complex, featuring multiple lines of text, as well as text rotation and translation. Furthermore, since the conventional BLEU (Papineni et al., 2002) is not applicable to image evaluation, we extend BLEU to Structure-BLEU that considers text location information to better evaluate the quality of text translations within images.

To summarize, we have the following major contributions in this work:

- We propose a novel end-to-end IIMT model named Translatotron-V. More importantly, it introduces two crucial modules to address major challenges in end-to-end IIMT: 1) target text decoder used to alleviate the modeling burden; 2) image tokenizer preventing the model from directly predicting pixels.
- We present a two-stage training framework for Translatotron-V, which fully exploits unlabeled images, OCR, and TIT data to refine the model training.
- We propose Structure-BLEU, an evaluation metric that considers text location information for IIMT.
- Experimental results demonstrate that Translatotron-V not only significantly outperforms the pixel-level end-to-end IIMT model, but also achieves comparable performance with fewer parameters to cascaded models.

## 2 Related Work

To achieve high-performance IIMT, previous research mainly focuses on text image translation (TIT), which is a subtask of IIMT (Watanabe et al., 1998; Yang et al., 2002; Du et al., 2011; Chen et al., 2015; Afli and Way, 2016; Lan et al., 2023). Unlike conventional multimodal machine translation (Elliott et al., 2016; Yin et al., 2020; Lin et al., 2020; Su et al., 2021a; Yin et al., 2023; Kang et al., 2023), TIT aims to translate source language texts in images into target language. In this regard, dominant studies resort to the cascading method, which uses an OCR model to obtain the recognized source lan-

guage texts and then feed them into an MT model for translation (Goodfellow et al., 2014; Zhang et al., 2016; Gu et al., 2018).

Afterwards, due to the advantages of mitigating error propagation, the end-to-end TIT attracts increasing attention. Chen et al. (2020) adopt multi-task learning framework that integrates OCR as an auxiliary task. Along this line, Ma et al. (2022) incorporating MT into the multi-task learning framework. Unlike previous studies, both Su et al. (2021b) and Ma et al. (2023b) employ an adapter to combine individual pretrained OCR and MT modules in a TIT model. Furthermore, Ma et al. (2023c) apply knowledge distillation to effectively distillate the knowledge of OCR and MT models into the end-to-end TIT model. Zhu et al. (2023) explore an end-to-end TIT model with an aligner and a regularizer to reduce the modality gap. To explicitly exploit guidance from recognized texts, Ma et al. (2023a) incorporate recognized text information into the TIT decoder through interactive attention. Differing from the above studies focusing on model design, Salesky et al. (2021) analyze the effect of visual text representation, and find that it exhibits significant robustness to various types of noise.

However, none of the aforementioned works consider generating the image with target translations, which is a common requirement in real-world scenarios. To this end, Mansimov et al. (2020) first explore the IIMT task. They introduce an end-to-end model that contains a self-attention encoder, two convolutional encoders, and a convolutional decoder to generate target images at the pixel level. Nonetheless, their model significantly lags behind cascaded models, suffering from issues such as character omission and artifacts. Recently, Tian et al. (2023) convert pixels into characters, thereby transforming the IIMT task into a conventional sequence-to-sequence text generation task. However, this method can only generate grayscale images, losing visual characteristics. Susladkar et al. (2023) present a conditional diffusion-based image editing model, which replaces text in the input image with a given translation while preserving the visual characteristics of origin image. However, this model can only perform single-word editing, which makes its application very limited.

Different from these studies, we propose an end-to-end IIMT model that can generate RGB images with multiple lines of text while preserving the vi-

sual features of the input image, and achieve comparable performance to the cascaded model.

### 3 Our Model

#### 3.1 Model Architecture

As shown in Figure 2, our model consists of four modules: an image encoder, a target text decoder, an image decoder, and an image tokenizer. All of those modules will be elaborated in the following.

**Image Encoder.** This module converts the input image into a sequence of visual vectors.

We use ViT (Dosovitskiy et al., 2021) as the backbone of the image encoder. In order to convert a 2D image into a 1D sequence that can be handled by Transformer, we first split the input image  $\mathbf{x}$  into  $N = HW/P^2$  image patches  $\{x_i\}_{i=1}^N$ , where  $(H, W)$  is the resolution of the input image, and  $(P, P)$  is the resolution of each patch. Then we apply a linear projection matrix  $\mathbf{W}_e$  to transform image patches into patch embeddings, and use a standard learnable positional embedding matrix  $\mathbf{E}_{pos}$  to further optimize these patch embeddings. Formally, the initial hidden states  $\mathbf{H}_{ie}^0$  of the image encoder can be formulated as

$$\mathbf{H}_{ie}^{(0)} = [x_0; \mathbf{W}_e x_1; \mathbf{W}_e x_2; \dots; \mathbf{W}_e x_N] + \mathbf{E}_{pos}, \quad (1)$$

where  $x_0$  is the special token prepended to the input sequence.

Afterwards, we process these patch embeddings using a Transformer encoder with multiple layers. Each Transformer encoder layer is composed of a self-attention sub-layer and a feed-forward network (FFN) sub-layer. Layernorm (LN) is applied before each sub-layer, and residual connections after each sub-layer (Wang et al., 2019). The hidden states  $\mathbf{H}_{ie}^{(l)}$  of the  $l$ -th encoder layer is calculated as

$$\mathbf{H}_{ie}^{(l)} = \text{FFN}(\text{MHA}(\mathbf{H}_{ie}^{(l-1)}, \mathbf{H}_{ie}^{(l-1)}, \mathbf{H}_{ie}^{(l-1)})), \quad (2)$$

where  $\text{MHA}(\cdot, \cdot, \cdot)$  denotes a multi-head attention function. The residual connection and layer normalization are omitted for simplicity.

**Target Text Decoder.** By utilizing the features generated by the image encoder, this decoder is responsible for producing text translations. In this way, it focuses on the alignment of different languages, and thus alleviates the modeling burden of the image decoder.

When constructing our target text decoder, we employ the widely-used Transformer (Vaswani et al., 2017) decoder as the architecture, consist-

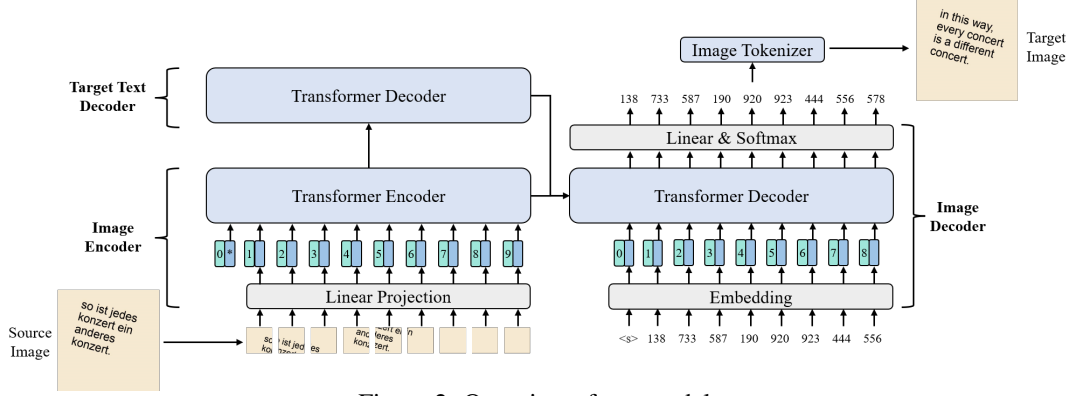


Figure 2: Overview of our model.

ing of multiple identical layers. In addition to the standard self-attention and FFN sub-layers, each decoder layer is equipped with a cross-attention sub-layer to exploit hidden states produced by the image encoder. Formally, we calculate the hidden states  $\mathbf{H}_{td}^{(l)}$  for the  $l$ -th decoder layer using the following equations:

$$\mathbf{C}_{td}^{(l)} = \text{MHA}(\mathbf{H}_{td}^{(l-1)}, \mathbf{H}_{td}^{(l-1)}, \mathbf{H}_{td}^{(l-1)}), \quad (3)$$

$$\mathbf{H}_{td}^{(l)} = \text{FFN}(\text{MHA}(\mathbf{C}_{td}^{(l)}, \mathbf{H}_{ie}^{(L)}, \mathbf{H}_{ie}^{(L)})), \quad (4)$$

where the initial hidden states  $\mathbf{H}_{td}^{(0)}$  are computed by summing the word embeddings and position embeddings of the input sequence. Unless otherwise specified, we use  $L$  to represent the last layer.

**Image Decoder.** This module is responsible for generating visual tokens based on visual and linguistic information generated from the image encoder and target text decoder, respectively.

The architecture of the image decoder closely resembles that of the target text decoder but with the following notable modifications. It includes two cross-attention sub-layers to gather information from both the image encoder and target text decoder, followed by a fusion sub-layer to generate intermediate representations enriched with both visual and linguistic features. Besides, we incorporate the 2D relative position encoding (Wu et al., 2021) into the self-attention sub-layer to capture relative positional relationships within images.

Let  $\mathbf{C}_{id}^{(l)}$  denote the hidden states output by the  $l$ -th self-attention sub-layer, we calculate it in the following way:

$$\mathbf{C}_{id}^{(l)} = \text{MHA}(\mathbf{H}_{id}^{(l-1)}, \mathbf{H}_{id}^{(l-1)}, \mathbf{H}_{id}^{(l-1)}), \quad (5)$$

where  $\mathbf{H}_{id}^{(l-1)}$  represents the hidden state output by the  $(l-1)$ -th image decoder layer. Subsequently, the hidden states  $\bar{\mathbf{H}}_{id}^{(l)}$  and  $\tilde{\mathbf{H}}_{id}^{(l)}$  are computed through two cross-attention mechanisms, which attend to

the image encoder and the target text decoder, respectively, as follows:

$$\bar{\mathbf{H}}_{id}^{(l)} = \text{MHA}(\mathbf{C}_{id}^{(l)}, \mathbf{H}_{ie}^{(L)}, \mathbf{H}_{ie}^{(L)}), \quad (6)$$

$$\tilde{\mathbf{H}}_{id}^{(l)} = \text{MHA}(\mathbf{C}_{id}^{(l)}, \mathbf{H}_{td}^{(L)}, \mathbf{H}_{td}^{(L)}). \quad (7)$$

Finally, the hidden states of the  $l$ -th image decoder layer are obtained through a gated fusion mechanism, which is calculated using the following equations:

$$\Lambda = \text{sigmoid}(\mathbf{W}_{\Lambda} \bar{\mathbf{H}}_{id}^{(l)} + \mathbf{U}_{\Lambda} \tilde{\mathbf{H}}_{id}^{(l)}), \quad (8)$$

$$\mathbf{H}_{id}^{(l)} = \Lambda \bar{\mathbf{H}}_{id}^{(l)} + (1 - \Lambda) \tilde{\mathbf{H}}_{id}^{(l)}, \quad (9)$$

where  $\mathbf{W}_{\Lambda}$  and  $\mathbf{U}_{\Lambda}$  are projection matrices, and  $\Lambda$  is a gated matrix featuring values ranging from 0 to 1, serving the purpose of dynamically fusing two modalities of information.

**Image Tokenizer.** It is used to perform the conversion between an image and a sequence of discrete visual tokens. By introducing this module, we allow the image decoder only to predict visual tokens, preventing it from modeling excessively lengthy sequences. For instance, a  $256 \times 256 \times 3$  RGB image results in 196,608 rasterized values.

Our image tokenizer follows the architecture of ViT-VQGAN (Yu et al., 2022a), which includes a Vision Transformer (ViT) (Dosovitskiy et al., 2021) based encoder and decoder. The encoder  $E$  of the image tokenizer is used to tokenize the image into  $\mathbf{z} = (z_1, \dots, z_N)$  through a quantizer  $q(\cdot)$ . Formally, the quantizer looks up the nearest visual token for each input, as shown in the following:

$$z_i = q(E(x_i)) = \underset{e_k \in \mathcal{V}}{\text{argmin}} \|E(x_i) - e_k\|_2, \quad (10)$$

where  $\mathcal{V}$  is the image vocabulary containing visual tokens.

Conversely, the decoder  $G$  of the image tokenizer reconstructs the input image based on the visual tokens generated by  $E$ , formulated as

$$\hat{\mathbf{x}} = G(q(E(\mathbf{x}))). \quad (11)$$

Please note that during training, we use the encoder to obtain visual tokens of the target image as labels. During inference, the decoder converts visual tokens generated by the image decoder into the target image.

### 3.2 Model Training

We provide a detailed description of the training procedures for our model, which consists of two stages, as illustrated in Figure 3.

**Stage 1.** At this stage, we train the image tokenizer using a large-scale unlabeled image dataset  $D_v$  in the same way as ViT-VQGAN (Yu et al., 2022a), where we convert the input image into visual tokens and then reconstruct the image from these visual tokens.

Given an image  $\mathbf{x}$  from the unlabeled image dataset  $D_v$ , we define the training objective of this stage as follows:

$$\mathcal{L}_1 = \|\hat{\mathbf{x}} - \mathbf{x}\|^2 + \|\text{sg}(E(\mathbf{x})) - \mathbf{z}\|_2^2 + \beta \|E(\mathbf{x}) - \text{sg}(\mathbf{z})\|_2^2. \quad (12)$$

Here, the first item is the reconstruction loss optimizing the encoder and decoder, the middle item is the vector-quantization loss used to update the visual tokens, the last item is the so-called ‘‘commitment loss’’ for the encoder which prevents its output fluctuating frequently from one visual token to another,  $\text{sg}(\cdot)$  denotes the stop-gradient operation, and  $\beta$  is the weighting factor set to 0.25 following van den Oord et al. (2017).<sup>2</sup>

**Stage 2.** Using an IIMT dataset, we then adopt multi-task learning and knowledge distillation to train the image encoder, target text decoder, and image decoder.

Overall, the training objective at this stage is defined as follows:

$$\mathcal{L}_2 = \mathcal{L}_{iimt} + \mathcal{L}_{ocr} + \mathcal{L}_{tit} + \mathcal{L}_{kd}. \quad (13)$$

where  $\mathcal{L}_{iimt}$ ,  $\mathcal{L}_{ocr}$ ,  $\mathcal{L}_{tit}$ , and  $\mathcal{L}_{kd}$  denote the IIMT task loss, OCR auxiliary task loss, TIT auxiliary task loss, and knowledge distillation loss, respectively.<sup>3</sup>

Given an IIMT training instance  $(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{t})$  from the IIMT dataset  $D_{iimt}$ , we can utilize the image tokenizer trained in the first stage to process the target image, obtaining visual tokens denoted as  $\mathbf{z}$ .

<sup>2</sup>Note that we also include other loss terms as presented in (Yu et al., 2022a), but omit the descriptions for brevity. Please refer to (Yu et al., 2022a) for more details.

<sup>3</sup>We also explore the balance of different training objectives. Experimental results in Appendix A show that we do not need to introduce additional hyperparameters to balance different objectives.

Here,  $\mathbf{x}$  represents the source image,  $\mathbf{y}$  is the target image,  $\mathbf{s}$  denotes the source language text within the source image, and  $\mathbf{t}$  denotes the target language text within the target image.

To alleviate the burden of end-to-end model training, we adopt multi-task learning, which involves not only the primary IIMT task but also two auxiliary tasks: the OCR task and the TIT task. The OCR task is employed to assist the model in recognizing texts within the image, while the TIT task further facilitates cross-lingual alignment. Formally, the training objective of the IIMT task can be formulated as follows:

$$\mathcal{L}_{iimt} = -\log p(\mathbf{z}|\mathbf{x}; \theta_{ie}, \theta_{ttd}, \theta_{id}), \quad (14)$$

where  $\theta_{ie}$ ,  $\theta_{ttd}$ ,  $\theta_{id}$  denote the trainable parameters of the **image encoder**, **target text decoder**, and **image decoder**, respectively.

To train our model using the OCR auxiliary task, we additionally introduce a source text decoder, which adopts the same architecture as the target text decoder. Formally, the training objectives of the OCR and TIT auxiliary tasks are defined as

$$\mathcal{L}_{ocr} = -\log p(\mathbf{s}|\mathbf{x}; \theta_{ie}, \theta_{std}), \quad (15)$$

$$\mathcal{L}_{tit} = -\log p(\mathbf{t}|\mathbf{x}; \theta_{ie}, \theta_{ttd}), \quad (16)$$

where  $\theta_{std}$  is the parameters of the **source text decoder**. Note that the source text decoder takes the intermediate hidden states of the image encoder as input. This design is based on the intuition that the shallow encoder layers represent the source visual content, while the deep layers encode more information about the target visual content.

Besides, training an end-to-end model is considerably more difficult than a T2I model, where the latter only needs to learn the mapping between different modalities and thus has better performance. Consequently, we introduce a T2I model as a teacher to facilitate knowledge transfer to the end-to-end model. This T2I model includes a Transformer-based text encoder, a ResNet-based image encoder (He et al., 2016), and an image decoder similar to our model, where the image encoder is used to preserve the features of the original image. Denote the output distribution of the teacher model for  $t$ -th visual token  $z_t$  as  $q(z_t|\mathbf{z}_{<t}, \mathbf{x}, \mathbf{t}; \theta_{t2i})$ , we define the cross-entropy between the distributions of teacher and student as the distillation loss:

$$\mathcal{L}_{kd} = - \sum_{t=1}^{|\mathbf{z}|} \sum_{k=1}^{|\mathcal{V}|} q(z_t = k|\mathbf{z}_{<t}, \mathbf{x}, \mathbf{t}; \theta_{t2i}) \log p(z_t = k|\mathbf{z}_{<t}, \mathbf{x}; \theta_{ie}, \theta_{ttd}, \theta_{id}), \quad (17)$$

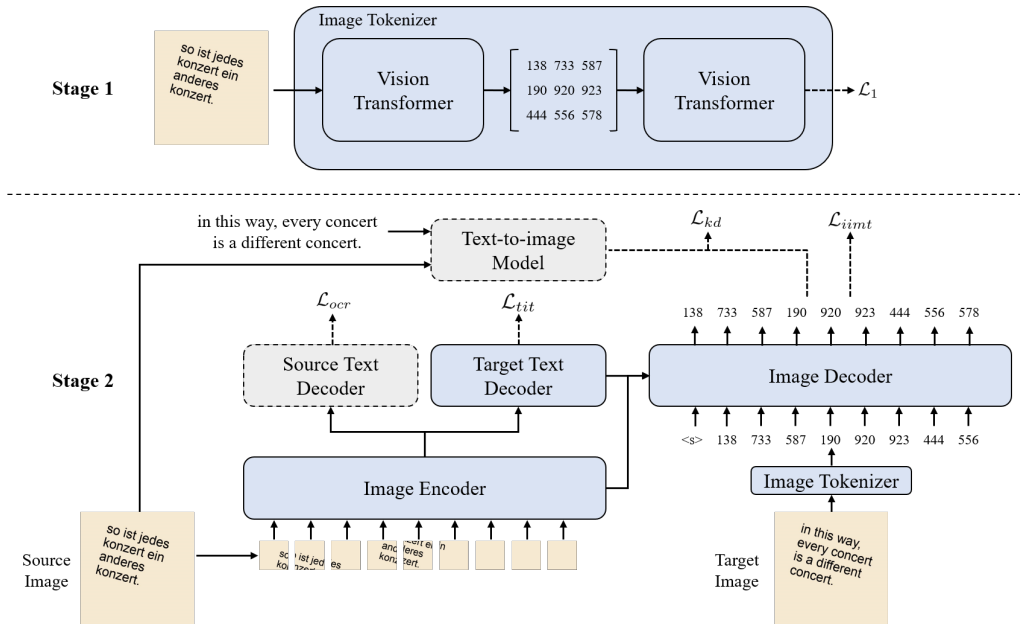


Figure 3: Overview of our two-stage training framework. The modules enclosed by dotted lines will be removed during inference.

where  $\theta_{t2i}$  represents the parameters of the T2I model. Note that we will remove the source text decoder and T2I model during inference.

## 4 Experiments

### 4.1 Setup

**Dataset.** Due to the lack of readily available data, we utilize the widely-used IWSLT14 German-English (De-En) dataset (Cettolo et al., 2014) to synthesize paired images for this task. Concretely, we leverage the Python Pillow package<sup>4</sup> to render texts onto images with the black Arial<sup>5</sup> font. The text is arranged horizontally from left to right, and vertically from top to bottom, with randomly translating and rotating. This involves shifting the text in a random direction and changing its orientation by a random angle. Additionally, the background color of the image is selected randomly and the resolution of the images is  $512 \times 512$ . Note that bilingual texts exceeding the image boundaries will be disregarded during the process of data synthesis. In contrast to prior studies (Mansimov et al., 2020; Tian et al., 2023), which focus solely on generating images with single-line text and white background, our research delves into more complex scenes. In the end, the synthesized dataset comprises 81,741 training instances, 3,765 validation instances, and 3,527 test instances. Several synthetic examples

<sup>4</sup><https://github.com/python-pillow/Pillow>

<sup>5</sup><https://learn.microsoft.com/en-us/typography/font-list/arial>

and comparisons with previous data can be found in Appendix B.

In addition to the IIMT data, a substantial quantity of images is also indispensable for training the image tokenizer. To this end, we employ the text extracted from the WMT14 English-German (En-De) (Bojar et al., 2014) to synthesize images for training our image tokenizer.

**Implementation Details.** In this work, we employ the same setting as ViT-B (Dosovitskiy et al., 2021) to construct our image encoder. Both our target text decoder and image decoder are composed of 8 layers, each of which has 512-dimensional hidden states, 8 attention heads, and 2,048 feed-forward hidden states. Besides, our image tokenizer is similar to ViT-VQGAN-SS (Yu et al., 2022a) but uses a smaller setup. It consists of 4 layers of encoder and decoder, each of which has 256-dimensional hidden states, 8 attention heads, and 1,024 feed-forward hidden states. Particularly, we use a character-level vocabulary of size 256 for the OCR and TIT auxiliary tasks, while the image vocabulary size for visual tokens is set to 8,192. Unless otherwise specified, the patch size of the image is set to 16.

During the first training stage, we train the image tokenizer with a batch size of 512 for 10,000 steps, where the parameters are updated by AdamW (Loshchilov and Hutter, 2019) with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.99$ . During the second stage, we train the model for 100 epochs with an early stopping patience

Model	De→En			En→De			#Param
	BLEU ↑	Structure-BLEU ↑	SSIM ↑	BLEU ↑	Structure-BLEU ↑	SSIM ↑	
<i>Cascaded Models</i>							
OCR+MT+T2I	15.37	14.87	0.7785	13.22	12.57	0.7550	247M
TIT+T2I	14.80	14.73	0.7812	12.92	12.74	0.7620	201M
PEIT+T2I	10.91	10.78	0.7740	8.77	8.01	0.7594	178M
<i>End-to-end Models</i>							
Pixel-level Transformer	0.15	0.15	0.7538	1.11	1.22	0.7616	162M
Translatotron-V	<b>15.39</b>	<b>15.26</b>	<b>0.7832</b>	<b>13.23</b>	<b>12.92</b>	<b>0.7629</b>	175M

Table 1: Experimental results on the De→En and En→De IIMT tasks.

Model	BLEU ↑	S-BLEU ↑	SSIM ↑
Translatotron-V	<b>15.39</b>	<b>15.26</b>	<b>0.7832</b>
w/o gated fusion	14.34	14.20	0.7830
w/o OCR auxiliary task	1.39	1.18	0.7277
w/o knowledge distillation	13.35	13.43	0.7813
w/o target text decoder	0.47	0.43	0.7751

Table 2: Ablation study of Translatotron-V on the De→En IIMT task. S-BLEU represents Structure-BLEU.

set to 10, and a batch size set to 80. This stage of training also utilizes the AdamW optimizer ( $\beta_1 = 0.9, \beta_2 = 0.999$ ) along with weight decay of 0.001 and polynomial decay learning rate scheduling. To alleviate overfitting, we apply a dropout rate of 0.1 and incorporate the label smoothing with a coefficient of 0.1, and we average the checkpoints of the last 10 epochs for evaluation.

**Baselines.** We construct the following baselines: OCR+MT+T2I, TIT+T2I, PEIT+T2I, and pixel-level Transformer, all models trained using character inputs and outputs similar to our model.

1) *OCR+MT+T2I*. This baseline cascades three models: an OCR model, an MT model, and a T2I model. Note that our teacher model has the same architecture as the T2I Model, except for reducing the hidden states from 512-dimensional to 384-dimensional. 2) *TIT+T2I*. We construct this baseline by cascading the TIT model and the T2I model. Additionally, it applies both OCR and TIT tasks during training to achieve better performance. 3) *PEIT+T2I*. This baseline is similar to TIT+T2I but replaces the multi-line TIT model with PEIT (Zhu et al., 2023), the state-of-the-art single-line TIT model. Since PEIT is designed for single-line TIT, during inference, we first employ the widely-used EasyOCR<sup>6</sup> as the detection model to recognize and crop each line of text from the image, and then concatenate them together into a single-line text image. 4) *Pixel-level Transformer*. This model

<sup>6</sup><https://github.com/JaidedAI/EasyOCR>

uses the same structure as our model but removes the image tokenizer and directly predicts pixel values. It is trained using multi-task learning as well, with the IIMT task being optimized with a mean squared error loss due to the pixel values being of floating-point type.

The detailed architecture of OCR+MT+T2I, TIT+T2I, and PEIT+T2I is described in Appendix C.

**Evaluation.** We evaluate the output images from both the perspectives of translation quality and image quality. We follow Mansimov et al. (2020) to transcribe the generated images into texts with EasyOCR toolkit and then measure the BLEU (Papineni et al., 2002) score calculated by SacreBLEU<sup>7</sup> (Post, 2018). To take into account the location of texts within the generated image, we extend the conventional BLEU to Structure-BLEU. This metric first performs OCR on the generated image and the reference image separately. Then, we use the bounding boxes in the generated image and the reference image to calculate intersection over union (IoU) for text matching. Subsequently, we filter out matched text pairs with significantly different positions, specifically those with IoU values below 0.5. Finally, we calculate the BLEU score for the remaining paired texts. For more comprehensive details, please refer to the algorithm provided in the Appendix D. Besides, we evaluate the quality of the generated images via structural similarity index measure (SSIM) (Wang et al., 2004), which considers luminance, contrast, and structure to measure the similarity between two images. The comparison between SSIM and BLEU can be found in Appendix E.

## 4.2 Results

Table 1 shows the overall results of all models on the IWSLT14 De-En datasets, and we have the following observations. First, OCR+MT+T2I exhibits

<sup>7</sup><https://github.com/mjpost/sacrebleu>

Source	OCR+MT+T2I	TIT+T2I	PEIT+T2I	Pixel-level Transformer	Translatotron-V	Reference
und was macht man dann mit dem müll?	and what do you do with the waste?	and what do you do with the waste that waste?	and what do you do, "you're like."	ano tmet; flo r mst yoruäthi mstlll marl'g testrlfo.	and then what do you do with the waste?	and then what do you do with the waste?

Figure 4: Translation examples of different models on the De→En IIMT task.

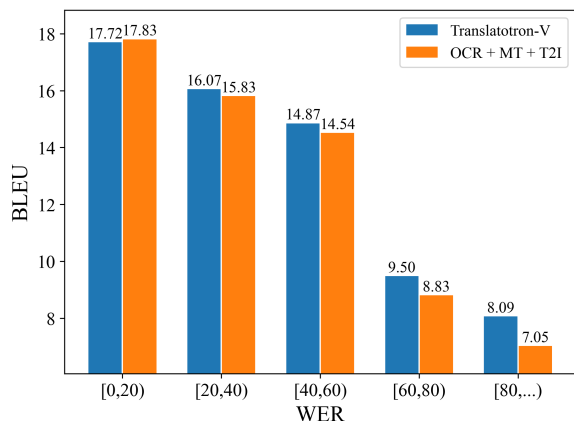


Figure 5: BLEU scores on different groups divided according to WER. We compute the WER using the text produced by the OCR toolkit.

significantly better translation quality compared to TIT+T2I, but its performance of image quality is relatively poor. The underlying reason is that cascading more models may lead to more severe error propagation, thereby affecting the image quality generated by the final T2I model. Second, PEIT+T2I performs worse than other cascaded models, primarily because it heavily relies on the text detection model when handling multi-line text images. Third, the image translation quality generated by Pixel-Transformer is extremely poor, and we observe that it is essentially incapable of producing readable text. This is due to the fact that the search space of image pixel values is extremely large, which poses a challenge to the model’s optimization in generating accurate pixel values. Finally, Translatotron-V consistently achieves competitive performance with 70.9% parameters compared to cascaded baselines and significantly outperforms the pixel-level Transformer.

### 4.3 Ablation Study

To explore the effectiveness of different components, we further compare Translatotron-V with its several variants, as shown in Table 2.

1) *w/o gated fusion*. In this variant, we remove

the gated fusion mechanism of the image decoder when performing cross-attention. Consequently, the image decoder sequentially performs cross-attention over the image encoder and the target text decoder to update hidden states. The result in Line 2 indicates that this change causes a decline in translation quality, suggesting that the gated fusion mechanism is useful for fusing information from two modalities.

2) *w/o OCR auxiliary task*. When constructing this variant, we remove the OCR auxiliary task during the model training. Upon analyzing Line 3, it becomes evident that this task can empower the model with the ability to perceive text within images, enabling the model to accomplish translation.

3) *w/o knowledge distillation*. We remove the knowledge distillation in this variant. As indicated in Line 4, there is a significant performance drop, which demonstrates that knowledge distillation effectively reduces the difficulty of training.

4) *w/o target text decoder*. In this variant, we remove the target text decoder from our model. The results reported in Line 5 demonstrate a drastic decline in performance. We can confirm that the end-to-end IIMT model imposes a substantial modeling burden. The target text decoder plays a pivotal role in mitigating the burden of achieving alignment between different languages.

### 4.4 Case Study

Figure 4 displays the translation results of different models on the De→En dataset. We can observe that Translatotron-V generates the correct target image, while the strongest baseline model OCR+MT+T2I missing partial strokes for the word “you” in the generated images. Besides, Pixel-level Transformer has issues like character omission and artifacts, making it unable to generate correct words. This result reveals that image tokenizer is important for the Translatotron-V.



Model	Fr→En			Ro→En			#Param
	BLEU ↑	Structure-BLEU ↑	SSIM ↑	BLEU ↑	Structure-BLEU ↑	SSIM ↑	
<i>Cascaded Models</i>							
OCR+MT+T2I	21.60	21.58	0.7738	18.34	18.61	0.7752	247M
TIT+T2I	21.87	21.78	0.7801	18.39	18.30	0.7764	201M
PEIT+T2I	18.51	18.55	0.7741	14.54	14.90	0.7704	178M
<i>End-to-end Models</i>							
Pixel-level Transformer	2.08	2.61	0.7753	1.58	2.11	0.7696	162M
Translatotron-V	<b>22.20</b>	<b>22.17</b>	<b>0.7811</b>	<b>18.44</b>	<b>18.73</b>	<b>0.7780</b>	175M

Table 3: Experimental results on the Fr→En and Ro→En IIMT tasks.

#### 4.5 The Effectiveness on Alleviating Error Propagation

To further investigate the impact of error propagation, we divide the test set of the De→En dataset into different groups based on the Word Error Rate (WER) of OCR. The higher WER indicates that the image is more difficult to deal with, where potential error propagation is more severe. As illustrated in Figure 5, the improvements of Translatotron-V over OCR+MT+T2I are more significant with the increase of WER. Thus, we confirm again that our end-to-end model has the potential advantage of alleviating error propagation.

#### 4.6 Evaluation on Other Language Pairs

In order to further validate the effectiveness of Translatotron-V, we conduct experiments on two distinct language pairs: French to English (Fr→En) and Romanian to English (Ro→En). We also use the previously-described data synthesis method to convert the IWSLT17 Fr-En and Ro-En datasets (Cettolo et al., 2017) into IIMT datasets. As shown in Table 3, Translatotron-V still achieves competitive performance compared to cascaded models and significantly outperforms the Pixel-level Transformer across different language pairs.

### 5 Conclusion

In this work, we have proposed Translatotron-V, which is the first end-to-end IIMT model capable of generating RGB images and achieving comparable performance to the cascaded model with only 70.9% of parameters. In addition to an image encoder and an image decoder, Translatotron-V is equipped with a target text decoder and an image tokenizer, which are used to alleviate the modeling burden and prevent the model from directly predicting pixels, respectively. Moreover, we present a two-stage training framework to assist the model in learning alignment across modalities

and languages. Furthermore, we introduce an evaluation metric, Structure-BLEU, which considers text location information to evaluate the quality of translations within the image. Experimental results demonstrate the effectiveness of our proposed model and training framework.

In the future, we are interested in training models using only parallel images, which is important when texts within the image are not available.

#### Limitations

Currently, the quality of generated target images depends on the quality of the image tokenizer. However, in our experiments, we find that it sometimes generates incorrect words, which may be due to its training using only images without explicitly considering linguistic information. Meanwhile, Translatotron-V does not exhibit a speed advantage over the cascaded model. This is due to the reason that visual token sequences are still much longer than text sequences, and both cascaded and end-to-end models need to spend most of their time decoding the image. A promising direction is to find coarser visual tokens with a shorter sequence length without degrading the quality of the generated images. Furthermore, the synthetic dataset is still not realistic enough. However, acquiring IIMT data from the real world is very challenging, how to create a more realistic IIMT dataset is also an important direction.

#### Acknowledgments

The project was supported by National Natural Science Foundation of China (No. 62036004, No. 62276219), and the Public Technology Service Platform Project of Xiamen (No. 3502Z20231043). We also thank the reviewers for their insightful comments.

## References

- Haithem Afli and Andy Way. 2016. Integrating optical character recognition and machine translation of historical documents. In *COLING workshop*.
- Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, and Lucia Specia, editors. 2014. *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Baltimore, Maryland, USA.
- Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. 2017. Overview of the IWSLT 2017 evaluation campaign. In *IWSLT*.
- Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th IWSLT evaluation campaign. In *IWSLT*.
- Jinying Chen, Huaigu Cao, and Premkumar Natarajan. 2015. Integrating natural language processing with image document analysis: what we learned from two real-world applications. *Int. J. Document Anal. Recognit.*
- Zhuo Chen, Fei Yin, Xu-Yao Zhang, Qing Yang, and Cheng-Lin Liu. 2020. Cross-lingual text image recognition via multi-task sequence to sequence learning. In *ICPR*.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.
- Jun Du, Qiang Huo, Lei Sun, and Jian Sun. 2011. Snap and translate using windows phone. In *ICDAR*.
- Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *ACL*.
- Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay D. Shet. 2014. Multi-digit number recognition from street view imagery using deep convolutional neural networks. In *ICLR*.
- Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *ICLR*.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.
- Ye Jia, Ron J. Weiss, Fadi Biadsy, Wolfgang Macherey, Melvin Johnson, Zhifeng Chen, and Yonghui Wu. 2019. Direct speech-to-speech translation with a sequence-to-sequence model. In *Interspeech*.
- Liyan Kang, Luyang Huang, Ningxin Peng, Peihao Zhu, Zewei Sun, Shanbo Cheng, Mingxuan Wang, Degen Huang, and Jinsong Su. 2023. Bigvideo: A large-scale video subtitle translation dataset for multimodal machine translation. In *ACL Findings*.
- 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 *ACL*.
- Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei A. F. Florêncio, Cha Zhang, Zhoujun Li, and Furu Wei. 2023. Trocr: Transformer-based optical character recognition with pre-trained models. In *AAAI*.
- Huan Lin, Fandong Meng, Jinsong Su, Yongjing Yin, Zhengyuan Yang, Yubin Ge, Jie Zhou, and Jiebo Luo. 2020. Dynamic context-guided capsule network for multimodal machine translation. In *ACMMM*, pages 1320–1329.
- Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.
- 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 *ICPR*.
- Cong Ma, Yaping Zhang, Mei Tu, Yang Zhao, Yu Zhou, and Chengqing Zong. 2023a. CCIM: cross-modal cross-lingual interactive image translation. In *EMNLP Findings*.
- Cong Ma, Yaping Zhang, Mei Tu, Yang Zhao, Yu Zhou, and Chengqing Zong. 2023b. E2TMT: efficient and effective modal adapter for text image machine translation. In *ICDAR*.
- Cong Ma, Yaping Zhang, Mei Tu, Yang Zhao, Yu Zhou, and Chengqing Zong. 2023c. Multi-teacher knowledge distillation for end-to-end text image machine translation. In *ICDAR*.
- Elman Mansimov, Mitchell Stern, Mia Xu Chen, Orhan Firat, Jakob Uszkoreit, and Puneet Jain. 2020. Towards end-to-end in-image neural machine translation. *CoRR*, abs/2010.10648.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.
- Matt Post. 2018. A call for clarity in reporting BLEU scores. In *WMT*.
- Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *ICML*.
- Elizabeth Salesky, David Etter, and Matt Post. 2021. Robust open-vocabulary translation from visual text representations. In *EMNLP*.

Jinsong Su, Jinchang Chen, Hui Jiang, Chulun Zhou, Huan Lin, Yubin Ge, Qingqiang Wu, and Yongxuan Lai. 2021a. Multi-modal neural machine translation with deep semantic interactions. *Inf. Sci.*

Tonghua Su, Shuchen Liu, and Shengjie Zhou. 2021b. Rtnet: An end-to-end method for handwritten text image translation. In *ICDAR*.

Onkar Susladkar, Prajwal Gatti, and Anand Mishra. 2023. Towards scene-text to scene-text translation. *CoRR*, abs/2308.03024.

Yanzhi Tian, Xiang Li, Zeming Liu, Yuhang Guo, and Bin Wang. 2023. In-image neural machine translation with segmented pixel sequence-to-sequence model. In *EMNLP Findings*.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *NeurIPS*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *ACL*.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*

Yasuhiko Watanabe, Yoshihiro Okada, Yeun-Bae Kim, and Tetsuya Takeda. 1998. Translation camera. In *ICPR*.

Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. 2021. Rethinking and improving relative position encoding for vision transformer. In *ICCV*.

Jie Yang, Xilin Chen, Jing Zhang, Ying Zhang, and Alex Waibel. 2002. Automatic detection and translation of text from natural scenes. In *ICASSP*.

Yongjing Yin, Fandong Meng, Jinsong Su, Chulun Zhou, Zhengyuan Yang, Jie Zhou, and Jiebo Luo. 2020. A novel graph-based multi-modal fusion encoder for neural machine translation. In *ACL*.

Yongjing Yin, Jiali Zeng, Jinsong Su, Chulun Zhou, Fandong Meng, Jie Zhou, Degen Huang, and Jiebo Luo. 2023. Multi-modal graph contrastive encoding for neural machine translation. *Artif. Intell.*

Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. 2022a. Vector-quantized image modeling with improved VQGAN. In *ICLR*.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. 2022b. Scaling autoregressive models for content-rich text-to-image generation. *Trans. Mach. Learn. Res.*

Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. 2016. Variational neural machine translation. In *EMNLP*.

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 *ACL*.

## A The effectiveness of the training objective with balancing coefficients

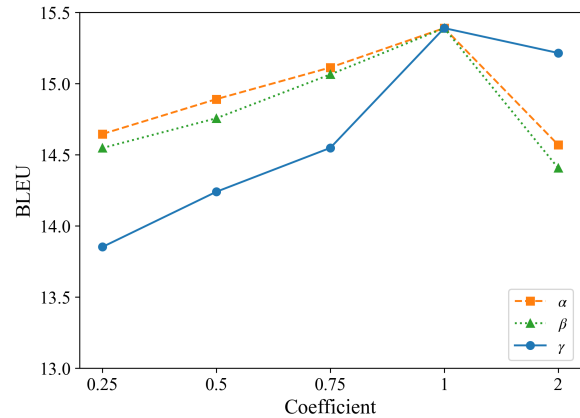


Figure 6: BLEU scores with different loss function coefficients.

To explore the impact of different weights of auxiliary task losses on model performance, we modify the training objective at the second stage as follows:

$$\mathcal{L}_2 = \mathcal{L}_{iimt} + \alpha\mathcal{L}_{ocr} + \beta\mathcal{L}_{tit} + \gamma\mathcal{L}_{kd}. \quad (18)$$

where  $\alpha$ ,  $\beta$ , and  $\gamma$  are the coefficient to control OCR auxiliary task loss  $\mathcal{L}_{ocr}$ , TIT auxiliary task loss  $\mathcal{L}_{tit}$ , and knowledge distillation loss  $\mathcal{L}_{kd}$ , respectively.

Due to the high cost of grid search, when adjusting a specific coefficient, all other coefficients will be set to 1. As shown in Figure 6, when all coefficients are set to 1, the model performs optimally. This result suggests that these tasks may be equally important and highly correlated. Adjusting a particular coefficient could lead the model to focus more on or neglect one task, which might not be beneficial if all tasks are equally important. This also

implies that our approach does not require carefully adjusting the coefficients for different training objectives.

## B Data Examples

In Figure 7, we present several data examples for our synthetic data. We also show the difference between our IIMT data and previous IIMT data in Figure 8. It can be observed that our data is more complex than the data used in previous work.

## C Baseline Architecture Details

In this section, we provide a detailed description of the architectures of two baselines: OCR+MT+T2I and TIT+T2I.

*OCR+MT+T2I.* This baseline cascades three models: an OCR model, an MT model, and a T2I model. First, we follow Li et al. (2023) to construct the OCR model, which consists of a ViT-B encoder and a Transformer decoder. The decoder uses the settings of Transformer Base, which contains 6 layers, 8 attention heads, 512-dimensional hidden states, and 2048 feed-forward hidden states. Second, we use the standard Transformer base (He et al., 2016) as the architecture of the MT model. Third, the T2I Model includes a Transformer-based text encoder, a ResNet-based image encoder (He et al., 2016), an image decoder, and an image tokenizer. Both the text encoder and image decoder utilize the same settings as Transformer base. Additionally, the image encoder adopts the ResNet50 architecture, and the image tokenizer follows the configuration of our model. It’s worth noting that our teacher model has the same structure as this T2I Model, except for reducing the hidden states from 512-dimensional to 384-dimensional.

*TIT+T2I.* We construct this baseline by cascading the TIT model and the T2I model, where the TIT model consists of an image encoder, a source text decoder, and a target text decoder. The image encoder uses the same architecture as ViT-B, and the configuration of the source text decoder and target text decoder is consistent with Transformer Base.

*PEIT+T2I.* This baseline replaces the multi-line TIT model in TIT+T2I with PEIT (Zhu et al., 2023). To ensure a fair comparison, we reimplement the model, scale its size to match our model, and do not use additional data during its multi-stage training.

## D Structure-BLEU

In Algorithm 1, we provide the detailed calculation process of Structure-BLEU. Furthermore, as shown in Figure 9, we also provide an example to demonstrate the difference between Structure-BLEU and BLEU. We can observe that Structure-BLEU takes into account the positional information of the text in the generated image, which is crucial for user experience in real-world scenarios.

---

### Algorithm 1 Structure-BLEU

---

**Require:** Generated image  $\hat{t}$ , reference image  $t$

- 1:  $\mathcal{R} \leftarrow \text{OCR}(t)$  {Set of reference texts and bounding boxes}
- 2:  $\mathcal{H} \leftarrow \text{OCR}(\hat{t})$  {Set of generated texts and bounding boxes}
- 3:  $\mathcal{M} \leftarrow \{\}$  {Set of matched text pairs}
- 4: **for** each  $h \in \mathcal{H}$  **do**
- 5:    $m \leftarrow \text{None}$
- 6:    $s \leftarrow 0$
- 7:   **for** each  $r \in \mathcal{R}$  **do**
- 8:      $\hat{s} \leftarrow \text{IOU}(h, r)$
- 9:     **if**  $\hat{s} > s$  **then**
- 10:        $s \leftarrow \hat{s}$
- 11:        $m \leftarrow (h, r)$
- 12:     **end if**
- 13:   **end for**
- 14:   **if**  $s \geq 0.5$  **then**
- 15:      $\mathcal{M} \leftarrow \mathcal{M} \cup \{m\}$
- 16:   **end if**
- 17: **end for**
- 18: **return**  $\text{BLEU}(\mathcal{M})$

---

## E Comparison of BLEU and SSIM

In this section, we provide examples illustrating the differences in focus between SSIM and BLEU evaluations. As shown in Figure 10, even when the text in the generated image is substantially incorrect, SSIM still yields a relatively high score. The underlying reason is that SSIM is used to evaluate the visual similarity between the generated image and the reference image, considering aspects such as brightness, contrast, etc. It focuses not only on the image of text regions but also on the image of non-text regions. However, BLEU is employed to evaluate the fine-grained information in the image of text regions, such as the glyph of the character. The two metrics have different focuses and complement each other. Therefore, the model may perform well in generating visual information, while it may

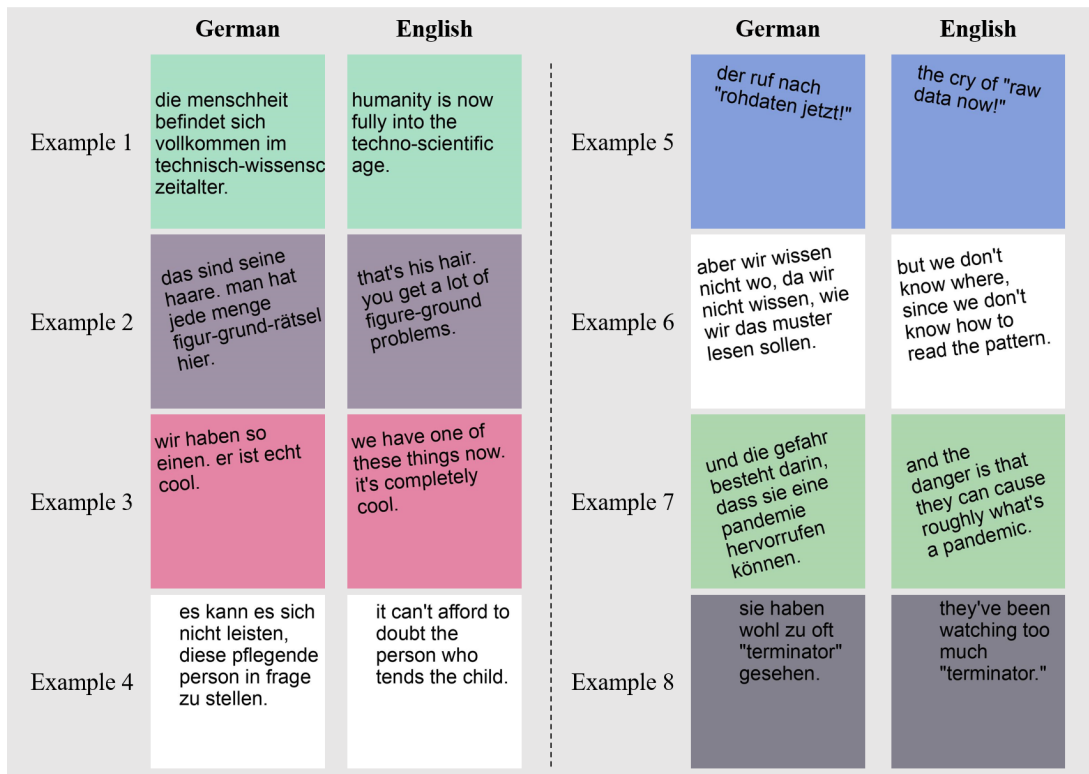


Figure 7: Several examples of our synthetic data using IWSLT14 De-En.

Dataset	Input Image	Output Image	Multiline	RGB Background	Rotation	Translation
Previous			×	×	×	×
Ours			✓	✓	✓	✓

Figure 8: Comparison of our IIMT data with other IIMT data used in previous work (Mansimov et al., 2020; Tian et al., 2023).

struggle with generating fine-grained text details, which will lead to a significant difference in BLEU scores but relatively close in SSIM scores.

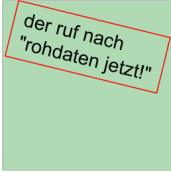
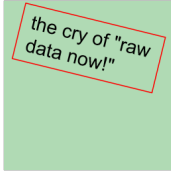

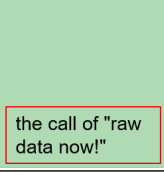
Input	Reference	Generated Image	Structure-BLEU	BLEU
			75.06	75.06
			0.0	75.06

Figure 9: Comparison of Structure-BLEU and BLEU. The bounding box of the text is drawn with red lines.

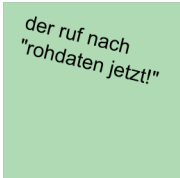
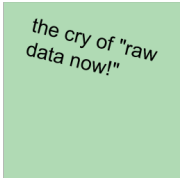


Input	Reference	Generated Image	BLEU	SSIM
			75.06	0.8074
			4.77	0.8927

Figure 10: Comparison of BLEU and SSIM.