Adding multimodal capabilities to a text-only translation model

Vipin Vijayan vipin255@gmail.com
Braeden Bowen bowen_braeden@bah.com
Scott Grigsby grigsby_scott@bah.com
PAR Government Systems Corporation, Dayton, OH*

Timothy Anderson timothy.anderson.20@us.af.mil
Jeremy Gwinnup jeremy.gwinnup.1@us.af.mil
Air Force Research Laboratory 711HPW/RHWTE, Dayton, OH

Abstract

While most current work in multimodal machine translation (MMT) uses the Multi30k dataset for training and evaluation, we find that the resulting models overfit to the Multi30k dataset to an extreme degree. Consequently, these models perform very badly when evaluated against typical text-only testing sets such as the newstest datasets.

In order to perform well on both Multi30k and typical text-only datasets, we use a performant text-only machine translation (MT) model as the starting point of our MMT model. We add vision-text adapter layers connected via gating mechanisms to the MT model, and incrementally transform the MT model into an MMT model by 1) pre-training using vision-based masking of the source text and 2) fine-tuning on Multi30k.

We achieve a state-of-the-art performance on the Multi30k 2016 en-de test set of 46.5 BLEU4 score and 0.61 CoMMuTE score via this approach while retaining the performance of the original text-only MT model against the newstest dataset.

1 Introduction

The task of multimodal machine translation (MMT) is to automatically translate text while using additional modalities (e.g., image, video, audio) to aid in translation. Prior work has shown that MMT can use contextually relevant images to aid in translation of sentences that contain ambiguities or missing textual information (Caglayan et al., 2019; Wu et al., 2021). For example, the noun "bank" is ambiguous and contextually dependent in English ("financial institution" or "river edge") but unambiguous in French ("*banque*" or "*rive*"). The hypothesis that these ambiguities or missing information can be resolved with contextually relevant images is persuasive.

Much work in MMT (Yao and Wan, 2020; Yin et al., 2020; Wu et al., 2021; Li et al., 2022) focus on the Multi30k dataset (Elliott et al., 2016), a dataset comprising 30,014 image captions and corresponding translations in different languages.

However, compared to the domain of text-only translation where MT models are trained using millions of examples, the Multi30k dataset is an extremely small dataset. Consequently, the MMT models will naturally overfit to the Multi30k dataset and perform poorly against testing sets that text-only translation models are typically evaluated against (Section 4).

Text-only machine translation is a much larger domain than multimodal machine translation and many strong models have been developed in the field (Kocmi et al., 2022). Thus, using a pre-trained text-only model as a starting point for MMT is a promis-

* Now doing business as Booz Allen Hamilton Corporation.

Proceedings of the 16th Conference of the Association for Machine Translation in the Americas,
Chicago, USA, September 30 - October 2, 2024. Volume 1: Research Papers

ing approach to advance the state of MMT. To demonstrate this, we incrementally transform a text-only MT model into an MMT model, resulting in state-of-the-art performance against the Multi30k dataset while retaining the performance of the pre-trained model against text-only test sets.

We use a pre-trained Transformer-based translation model as our starting point. We evolve this text-only translation model into an MMT model using adapters (Houlsby et al., 2019) and gating mechanisms such that the model learns how to use visual information while preserving its original translation performance. We do this by 1) combining a strong pre-trained translation model and a pre-trained vision-language model to create an MMT model, 2) pre-training the MMT model on a dataset of captions augmented with informed visual grounding and machine generated translations along with a dataset collated from a text-only MT dataset, and 3) fine-tuning against the Multi30k dataset.

Using this model architecture and training process, we achieve high performance against the Multi30k test sets while retaining high performance against text-only testing sets (Table 1).

2 Related Works

2.1 Adapting pre-trained models for MMT

Caglayan et al. (2021) converted a translation language model into a vision-based translation language model by pre-training using Conceptual Captions (Sharma et al., 2018), translating English captions to German using a translation model, and fine-tuning using Multi30k.

Futeral et al. (2023) also proposed a model that adapts a language model into an MMT model by simultaneously training against the MMT objective using the Multi30k dataset and the visually-conditioned masked language modeling objective using the Conceptual Captions dataset. While they used a visual-conditioned masked language modeling object, we use the much simpler training process of directly optimizing the output using cross-entropy loss. Furthermore, while they randomly choose words for visual grounding, we choose vision-based words selected using an object detection method for our masking.

2.2 Masking for visual grounding

Masking words for visual grounding is a common approach employed by such works as Wu et al. (2021), Ive et al. (2019), Caglayan et al. (2019), Wang and Xiong (2021). We cover a subset of these works.

Ive et al. (2019) masked specific words (ambiguous, inaccurate, and gender-neutral words) in the English source text to force the MMT models to use the visual information to generate target texts. They show that the additional visual context was helpful in text generation.

Caglayan et al. (2019) performed masking based on color deprivation, whole entity masking, and progressive masking on source texts. However, they found that training based on masking results in performance degradation on the Multi30k testing sets, which indicates that the vision information was not being fully utilized by their models.

Wang and Xiong (2021) performed masking of source text based on Flickr30k-Entities (Plummer et al., 2016) that were vision related and used a multi-task object to train their MMT model, where they optimized for object-masking loss in addition to the text generation.

2.3 Gating mechanism for MMT

Similar to our work, Wu et al. (2021), Zhang et al. (2020), Lin et al. (2020) and Yin et al. (2020) use a trainable gating mechanism in the context of MMT to control the fusion between vision and text. However, our work uses two gating parameters each for the six adapter layers that we add, totaling 12 gating parameters, which is considerably fewer than in their work, which uses two trainable gating matrices of size 2048×512 and $T \times 512$ where T is the number of input text tokens. Furthermore, while the average of the gating parameters used by Wu et al. (2021) tended towards 0.0 (consequently weighing vision information lower) as more training is done, we show in this work how the use of vision-based masking allows the training of our gating mechanism to use more of the vision information.

3 Methods

We take a similar approach that Alayrac et al. (2022) used to create their generative vision-language model, Flamingo, while adapting their approach for the MMT task.

Flamingo is a generative decoder-only vision-language model created by combining a pre-trained generative language model and a pre-trained vision model, where vision and text interactions are mod-

eled by via gated vision-text cross-attention layers inserted before each decoder layer. Then, the model is incrementally converted from using only text information to using both vision and text information by freezing the pre-trained portions of the model. The gating values are set to 0.0 at the beginning of training in order that the vision-language model initially performs equivalently to the language model, and as training progresses the gating values diverge from 0 via back-propagation and consequence learns to use vision information gradually.

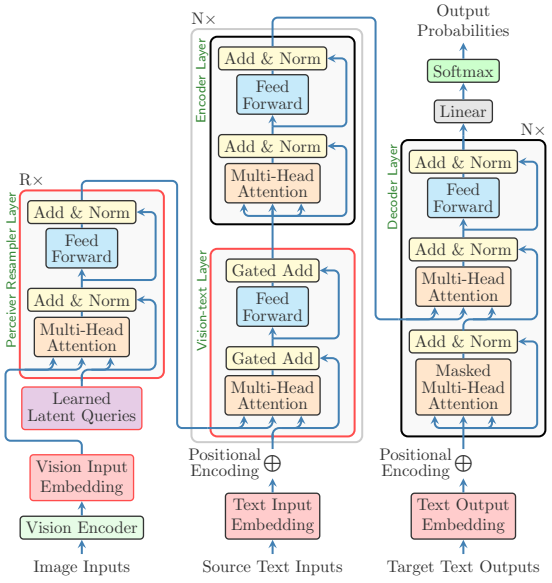

Figure 1: Multimodal translation architecture, where multimodal components are incorporated into the Transformer translation model introduced by Vaswani et al. (2017). The parameters in the model bordered by red are initialized randomly and updated for training, while the parameters in the pre-trained vision encoder and the pre-trained Transformer translation model bordered by black are frozen. The gating parameters in the vision-text layers are updated using back-propagation, allowing us to smoothly transition from a text-only translation model into a multimodal translation model.

Analogously, we start from a pre-trained Transformer-based text-only MT model and a pre-trained vision model to create an MMT model by inserting a vision-text cross-attention layer before each encoder layer. Using trainable gating parameters, we incrementally convert the model from using only text information to using both vision and text information to perform translation. We call our model GRAM (Gating and Residual Adapter-based Model).

While trainable gating parameters have been used in previous work for MMT (Wu et al., 2021; Zhang et al., 2020; Lin et al., 2020; Yin et al., 2020), our work is unique in the much lower number of gating parameters and in that it allows for the smooth transition of the model from performing as an MT model to performing as an MMT model.

Both the Flamingo model and our model were trained using the next-token prediction task, as is typical for text-only machine translation. Unlike Flamingo, which is a decoder-only model, our model is an encoder-decoder model. We inserted the vision-text layers before each of the encoder layers only, as we found it to perform better than inserting vision-text layers before the decoder layers only or before both the encoder and decoder layers (Appendix D.1). Aside from the perceiver resampler module and the gated vision-text cross attention layers used in Flamingo model, which we use to convert our model from an MT model to an MMT model, our GRAM model follows the original text-only Transformer MT model's hyper-parameters, layers, and training objectives as closely as possible.

3.1 GRAM model architecture

We start with a pre-trained Transformer translation model introduced by Vaswani et al. (2017) and add lightweight multimodal components (Figure 1). We use a pre-trained vision encoder, CLIP, to encode the input images (Section 3.1.1). We then link the vision encodings to the Transformer translation model using two components, the perceiver resampler (Section 3.1.2) and the vision-text layers (Section 3.1.3). The vision encodings, which can come from an arbitrary number of images, are converted into a fixed number of vision tokens using the perceiver resampler. Then, interactions between the vision tokens and the text embeddings are modeled using the vision-text cross-attention layers. The vision-text layers are incorporated into the Transformer layers by interleaving the vision-text layers and the original self-attention layers of the Transformer encoder.

In more detail, given an input sequence of text tokens $\boldsymbol{t} = (t_1, \ldots, t_n)$ and images $\boldsymbol{I} = (I_1, \ldots, I_l)$ where n and l may vary depending on the number

of input text tokens and images, the output token sequence is generated auto-regressively as follows.

The vision encoder maps the images $\boldsymbol{I}$ into vision encodings $\boldsymbol{v} = (v_1, \ldots, v_l)$ where $v_i \in \boldsymbol{R}^e$ and e is the size of the image encodings. The vision input embedding layer maps the vision encodings $\boldsymbol{v}$ into vision embeddings $\boldsymbol{w} = (w_1, \ldots, w_l)$ where $w_i \in \boldsymbol{R}^d$ and d is the size of the text and image embeddings. The text input embedding layer maps the text tokens $\boldsymbol{t}$ to text embeddings $\boldsymbol{x} = (x_1, \ldots, x_n)$ where $x_i \in \boldsymbol{R}^d$. The perceiver resampler remaps the variable number of image embeddings to a constant number of vision tokens $\boldsymbol{p} = (p_1, \ldots, p_r)$ where $p_i \in \boldsymbol{R}^d$, using the r learned latent queries.

Then, the encoder, consisting of a sequence of interleaved vision-text cross-attention layers and encoder layers, maps the text embeddings $\boldsymbol{x}$ and vision tokens $\boldsymbol{p}$ into a sequence of representations $\boldsymbol{z} = (z_1, \ldots, z_n)$ where $z_i \in \boldsymbol{R}^d$. Given $\boldsymbol{z}$, the decoder generates the output probabilities for the next output token in an auto-regressive manner, thus producing the output token sequence, $y_1, \ldots, y_m$.

3.1.1 Vision encoder

We use a pre-trained vision-language model, CLIP (Radford et al., 2021), to encode the input images. CLIP was trained on 400 million image-text pairs using a contrastive image-text approach. The vision encodings produced by CLIP contain rich semantic information relevant to vision-language tasks, and it has been shown to perform well on a wide variety of these tasks. We use the vision encoder in CLIP's best performing ViT-L/14@336px model, which outputs vector encodings of length 768.

3.1.2 Perceiver resampler

The perceiver resampler, used for the Flamingo model, receives a variable number of vision embeddings and outputs a fixed number of vision tokens. This concept was initially used to map a large number of inputs to a fixed number of tokens (Jaegle et al., 2021) and for object detection, where each of the visual tokens corresponds to an object class (Carion et al., 2020).

Given the vision embeddings $\boldsymbol{w}$, let $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_r)$ be the learned latent queries, and let **MHA** and **FF** be the multi-head attention layer and the feed forward layer, respectively. Then, the first perceiver resampler layer **PR** is $\mathbf{PR}(\boldsymbol{\lambda}, \boldsymbol{w}) = \boldsymbol{\lambda'} + \mathbf{FF}(\boldsymbol{\lambda'})$ where $\boldsymbol{\lambda'} = \boldsymbol{\lambda} + \mathbf{MHA}(K{=}[\boldsymbol{w}, \boldsymbol{\lambda}], V{=}[\boldsymbol{w}, \boldsymbol{\lambda}], Q{=}\boldsymbol{\lambda})$ and $[\boldsymbol{w}, \boldsymbol{\lambda}]$ is the concatenation of the two vectors. Then, the perceiver resampler layers continue with $\boldsymbol{\lambda} \leftarrow \mathbf{PR}(\boldsymbol{\lambda}, \boldsymbol{w})$ for R layers. The vision tokens $\boldsymbol{p} \leftarrow \boldsymbol{\lambda}$ are outputted by the final perceiver resampler layer.

3.1.3 Vision-text layer

Similar to the Flamingo model, in order to smoothly train our MMT model to ensure it behaves at the beginning of training like the pre-trained MT model and behaves at the end of training like an MMT model, we insert vision-text cross-attention layers before each of the original Transformer encoder layers and we use a gating mechanism for each of the vision-text layers.

Given the vision tokens $\boldsymbol{p}$ output by the perceiver resampler and the input text embeddings $\boldsymbol{x}$, let g_a and g_f be the learnable gating parameters for the multi-head attention layer **MHA** and the feed forward layer **FF** respectively, with $\gamma_a = \tanh(g_a)$, $\gamma_f = \tanh(g_f)$. Then, the first gated cross-attention layer **GCA** is $\mathbf{GCA}(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{x'} + \gamma_f \mathbf{FF}(\boldsymbol{x'})$ where $\boldsymbol{x'} = \boldsymbol{x} + \gamma_a \mathbf{MHA}(K{=}\boldsymbol{p}, V{=}\boldsymbol{p}, Q{=}\boldsymbol{x})$. The gated cross-attention layers then continue with $\boldsymbol{x} \leftarrow \mathbf{E}(\mathbf{GCA}(\boldsymbol{x}, \boldsymbol{\lambda}))$ for N layers where **E** is the original Transformer encoder layer following the cross-attention layer.

Gating parameters are set to 0.0 at the start of training and so it passes the text embeddings $\boldsymbol{x}$ through without modification. As training continues and as more vision information is used, $|g_a|$ and $|g_f|$ increases via back-propagation; consequently $|\gamma_a| = |\tanh(g_a)|$ and $|\gamma_f| = |\tanh(g_f)|$ approaches 1.0, since the tanh function maps the gating parameters g_a and g_f to be between -1.0 and 1.0

Since the gating parameters initially start at 0.0, vision information is ignored and the model performs as well as the text-only Transformer. During the training process the gating parameters are updated to gradually incorporate vision information for the multimodal translation task. The gating parameters can be seen as a proxy for how much vision information is used by the model.

3.1.4 Model hyper-parameters

During training, only the multimodal components are updated, while the vision encoder and the rest of the parameters in the text-only Transformer are kept frozen. For the vision encoder, we use pre-trained weights from the CLIP vision encoder model and

ignore CLIP's text encoder model[1]. For the text-only translation components, we use weights from the pre-trained MT model from FAIR's WMT19 submission[2]. Since our model uses FAIR's WMT19 MT model, we use the same byte-pair encoding (BPE) and vocabulary used by the MT model. Since the text-only portion of the model is frozen, training is relatively fast, typically 3 batches/sec using two Nvidia V100 GPUs where each batch contains 3,584 tokens.

Since we use the FAIR's WMT19 text-only Transformer as the starting point, we use those hyperparameters for our additional layers unless otherwise specified. For the perceiver resampler, we use two layers, i.e., $R = 2$, as was done for Flamingo. For both the perceiver resampler and the vision-text cross attention layers, we use the same parameters as in the text-only Transformer model, except for the number of attention heads being 16 and the intermediate feed-forward layer size being 4,096. The number of parameters are detailed in Appendix A).

3.2 Training

Beginning with the pre-trained text-only translation model, we add vision embedding layers and gated adapter layers that to the translation model to create a multimodal translation model (Section 3.1). Then, setting the initial gating values to 0.0, which allows our MMT model to perform equivalently to the MT model, we freeze the text-only parameters and train the additional vision-text parameters. We first pre-train the vision-text parameters of our model (Section 3.2.1) and then fine-tune the vision-text parameters using the Multi30k dataset (Section 3.2.2). During training, the gating value diverge from 0.0 as more vision information gets used by the model.

3.2.1 Pre-training

The intent of the pre-training step is to force the model to use contextually relevant image information by masking vision related words in the source sentence while performing the translation task. We pre-train our model on a dataset collated using vision-based masking of source sentences that we call the CR dataset.

First, we translate 2,878,999 of the English captions in the Conceptual Captions (CC) dataset (Sharma et al., 2018) that had images available to German using FAIR's WMT19 translation model, and then perform vision-based masking on the English captions.

For vision-based masking, we create a list of vision related phrases, or topic phrases, by using the VinVL object detector (Zhang et al., 2021) against the CC images. VinVL is able to detect 1,848 object classes and 524 attribute classes, resulting in a much richer possible vocabulary than other object detectors. With relatively high thresholds of 0.8 for object classes and 0.7 for attribute classes, we create a list of 7,494 "`attribute object`" combinations, such as "red car".

Then, for each English-German sentence pair, we search for topic phrases in the English sentence. For each topic phrase we find, we replaced it with the <unk> token (as we are restricted to using tokens present in the pre-trained FAIR WMT19 model, the <unk> token is the closest available token to a mask token). This results in an MMT dataset of 2,663,331 (masked source text, target text, image) triplets.

In addition, we also concatenate to the CR dataset 2,878,999 (<unk>, target text, image) triplets created from each of the captions in the CC dataset to further force the usage of vision information to generate text.

Furthermore, so that the model does not overfit to inputs that always contain image information while still maintaining the capacity to translate complex sentences, we concatenate to the CR dataset 1,183,301 (source text, target text, $\varnothing$) triplets created from the RAPID 2019 (Kocmi et al., 2022) dataset.

We train our GRAM model using the typical cross-entropy loss for machine translation. The optimization details for the pre-training step are described in Appendix C.1.

3.2.2 Training against Multi30k

Fine-tuning. We use the same vision-based masking described in Section 3.2.1 for the source sentences in the Multi30k training set, which resulted in 29,000 masked source text, target text, and image triplets. We refer to this resulting dataset as M30k. Since the Multi30k dataset contains only 29,000 examples, fine-tuning after the above pre-training step resulted in much better performance compared to directly

[1] See https://github.com/openai/CLIP to download weights.

[2] The weights are from the transformer.wmt19.de-en single model located in the pytorch/fairseq torch hub. See https://github.com/facebookresearch/fairseq for details.

training against the M30k dataset (Section 4).

Note that we train our model using a concatenation of the Multi30k training set with images and the Multi30k training set without images. This is to account for evaluation artifacts where the model performance when given both text and image input is higher than model performance with only text input, but the result is only due to the model overfitting on training data that only has (source text, target text, image) triplets and no examples of (source text, target text, $\varnothing$) triplets. We also explore fine-tuning of three other dataset variations including the original unmasked Multi30k dataset, which we discuss in Appendix E. The optimization details are described in Appendix C.2.

Direct training. We also directly training using the above described Multi30k dataset without the pre-training step for comparison. Due to its small size, we also explored directly training against the Multi30k dataset using smaller perceiver resampler and vision-text layers, and found performance to be similar (Appendix D.2). Thus, we show performance results using the same model sizes.

4 Results and Discussion

We use the evaluation framework proposed by Vijayan et al. (2024), where they argued that MMT models should be evaluated by measuring both 1) their use of visual information to aid in the translation task and 2) their ability to translate complex sentences as is done for text-only machine translation.

We evaluate model performances against 1) the CoMMuTE (Futeral et al., 2023) test set, 2) the Multi30k (Elliott et al., 2016) test sets, and 3) the WMT news translation task (Kocmi et al., 2022) test sets (newstest) using CoMMuTE score and BLEU4 calculated using SacreBLEU (Post, 2018).

The main evaluation results are shown in Table 1 and two examples from the CoMMuTE test dataset are shown in Figure 2. The label FAIR-WMT19 shows our model's performance before our training process, i.e., the original text-only Transformer's performance. M_{CR} is our GRAM model pre-trained on the CR dataset (Section 3.2.1). $M_{\text{CR,M30k}}$ is our model pre-trained on CR and fine-tuned on Multi30k (Section 3.2.2). M_{M30k} is our model trained on Multi30k without the pre-training step (Section 3.2.2). We compare against the Gated Fusion and RMMT models (Wu et al., 2021), which are both trained solely on the Multi30k dataset, as well as the reported performance of VGAMT (Futeral et al., 2023), which was introduced along with the CoMMuTE test set.

Label	CoMMuTE	Multi30k		newstest	
		2016	2017	2019	2020
	Score	BLEU4			
Multimodal inputs					
M_{CR}	0.57	39.2	36.8		
$M_{\text{CR,M30k}}$	**0.61**	46.5	**43.6**		
M_{M30k}	0.50	45.9	42.7		
Gated Fusion	0.50	42.0	33.6		
VGAMT	0.59	43.3	38.3		
Text inputs only					
FAIR-WMT19	0.50	40.7	37.7	40.6	**36.2**
M_{CR}	0.50	40.2	37.8	40.6	35.4
$M_{\text{CR,M30k}}$	0.50	46.4	42.9	**42.7**	**36.2**
M_{M30k}	0.50	45.9	42.8	36.1	26.8
RMMT	0.50	41.5	33.0	1.3	0.8
Non-matching inputs					
M_{CR}	0.51	39.0	36.7	42.1	35.6
$M_{\text{CR,M30k}}$	0.51	**46.6**	43.2	42.0	**36.2**
M_{M30k}	0.50	45.9	42.8	36.1	26.8
Gated Fusion	0.50	42.0	33.6	1.3	0.6

Table 1: Performance results for English to German (en-de) translations. The label FAIR-WMT19 shows our model's performance before our training process, i.e., the original text-only Transformer's performance. M_{CR} is our model pre-trained on the CR dataset; $M_{\text{CR,M30k}}$ is our model pre-trained on CR and fine-tuned on Multi30k; M_{M30k} is our model trained on Multi30k without the pre-training step; Gated Fusion and RMMT are our evaluations of the models published by Wu et al. (2021); VGAMT is the reported performance of the model published by Futeral et al. (2023). "Text inputs only" shows performance of when only the source text is given. "Multimodal inputs" shows the performances when both source text and image is used as input. "Non-matching inputs" shows performance when source text along with a random image is used as input.

4.1 Pre-training using vision-based masking

Since we begin with a performant MT model, we expect that our model will retain the high text-only performance of the MT model while transforming into an MMT model. In order to ensure this, we fol-

lowed the work by Alayrac et al. (2022), where they incrementally transformed a language model into a vision-language model which retaining text-only performance, both in terms of the design of our model architecture and our training process (Section 3).

Figure 2: Examples from the CoMMuTE test dataset of our model (the $M_{\text{CR,M30k}}$ model from Table 1) resolving ambiguous input text when given contextual images. The ambiguous words in the input sentences and the resolved ambiguities in the output and reference sentences are in *italics*.

Similar to Alayrac et al. (2022), we found that a pre-training step is necessary to successfully transform the model without performance loss. When we pre-train our model and then fine-tune against the Multi30k dataset, this results in state-of-the-art performance against the Multi30k test sets and CoMMuTE score (Table 1, label $M_{\text{CR,M30k}}$), as well as little to no degradation of performance against the newstest datasets.

However, when we train against the Multi30k dataset without pre-training, we achieve good performance in the Multi30k test sets but only 0.5 for the CoMMuTE score (Table 1, M_{M30k}), which indicates that image information is not being used by the model, and degraded performance on the newstest datasets (e.g., 36.2 BLEU4 on newstest2020 for the text-only FAIR-WMT19 model compared to 26.8 BLEU4 for M_{M30k}).

While our pre-training step does degrade performance slightly on the newstest datasets compared to the original text-only Transformer (e.g., 36.2 BLEU4 on newstest2020 for the text-only FAIR-WMT19 model compared to 35.4 BLEU4 for the M_{CR} model), we note that our pre-training process is relatively rudimentary (Section 3.2.1) while FAIR-WMT19 is a model that was fine-tuned specifically for the news translation task using the news commentary dataset (Ng et al., 2019). Interestingly, and contrary to expectations, fine-tuning on the Multi30k dataset after pre-training improves performance against the newstest2019 and newstest2020 datasets, which might indicate that the FAIR-WMT19 model is overfitted to the news commentary dataset.

4.2 Training against Multi30k without pre-training

Due to the small size of the Multi30k training set, it is expected that models trained against Multi30k without pre-training would perform badly against testing sets such as the newstest datasets. For comparison, in the text-only translation domain, MT models such as FAIR-WMT19 are trained on millions of examples and then evaluated against the newstest dataset. We evaluated the Gated Fusion and RMMT MMT models, introduced by Wu et al. (2021) and trained solely on Multi30k, against the newstest datasets. As expected, there is a drastic drop in performance when the models are evaluated against the newstest datasets (Table 1).

For the Gated Fusion model, we evaluate by associating random images to the source text and evaluate against the newstest datasets. Since the associated images are not necessarily related to the source text, this can be considered non-matching evaluation. For the RMMT model, which takes as input only the source text, and uses the source text to perform image retrieval for the translation task, we simply use the source text to evaluate against the newstest datasets. As shown in Table 1, while the models perform well against the Multi30k test sets, they perform very badly against the newstest datasets.

In contrast, since our model uses a performant text-only MT model as the starting point, our model performs well when given non-matching inputs while still having high performance against CoMMuTE and the Multi30k testsets.

4.3 Text-only translations in Multi30k

One point to note when evaluating against the Multi30k test sets is that most of its captions do not require the image in order to be correctly translated

due to the captions being unambiguous. Specifically, Futeral et al. (2023) analyzed the Multi30k Test2016 and Test2017 and showed that only 2.1% and 2.0%, respectively, of the examples in the test sets have ambiguous source sentences that can be resolved using the associated images. Thus, we expect that correct translations can be achieved with the text alone without the associated images for the vast majority of the remaining examples. Fitting our expectations, we see that state-of-the-art performance on the Multi30k test sets can be achieved without making use of image information at all (Table 1, "Text inputs only" rows).

Since high performance can be achieved on the Multi30k test sets without the use of contextual images, it is important that an evaluation framework such as the CoMMuTE evaluation framework that can confirm that visual information is being used to aid in the translation task should always be used in conjunction with the Multi30k test sets when evaluating MMT models.

4.4 Gating parameters

As in the Flamingo model, our model uses gating parameters to transform from a model that uses only text information to a model that uses both vision and text information to produce outputs. The gating parameters, explained in Section 3.1, can be viewed as how much the model weighs the image information compared to the text information. Since the g_f can potentially solely use text information in the training set, the g_a values should be interpreted as the main proxies that indicate how much image information influences the output of the model.

Gating parameters have been used previously for MMT, with Wu et al. (2021) having explored in detail how gating parameters that weighed vision and text information are affected in MMT models. For their model, as training progressed, the average value gating parameters tended towards 0.0, indicating that their model learned to not use image information as training progressed.

In contrast, our gating parameters did not trend towards 0.0 as training progressed (Figure 3), primarily due to the pre-training approach that we employ (as indicated by the difference in the progress of the gating values in pre-training vs. direct training in Figure 3). However, unlike in the Flamingo model, where the maximum of the attention gating values $|\gamma_a| = |\tanh g_a|$ reaches around 0.8 towards the end of training, and the maximum of the feed-forward gating values $|\gamma_f| = |\tanh g_f|$ reaches 0.95, our gating values reach 0.035 for $|\gamma_a|$ and 0.2 for $|\gamma_f|$. This suggests that image information is not necessarily as important for the multimodal translation task compared to the Flamingo model, which can perform a wide variety of tasks including visual question answering. On the other hand, improvements in the training datasets and processes may increase the gating values to be closer to that of the Flamingo model.

5 Conclusion

Text-only machine translation is a much larger domain than multimodal machine translation and many strong models have been developed in the field. The approach of transforming a language model into a vision-language model was successful demonstrated via Flamingo, and thus have a high probability of working well in the similar task of machine translation. Following this idea, we designed an MMT model that began as a performant text-only MT model and incrementally transformed it into a MMT model by 1) pre-training using informed vision-based masking of the source text and 2) fine-tuning on Multi30k. We achieved a state-of-the-art performance on the Multi30k 2016 test set of 46.5 BLEU4 score via this approach while retaining high performance against CoMMuTE and the newstest datasets. There are many approaches for improving our model including the training process, where the pre-training dataset can be improved using more text-only datasets or augmenting text-only datasets using image retrieval, and model architecture, where techniques such as VMLM can be used to further enforce the use of image information in the model.

Acknowledgements

Thanks to AFRL SCREAM Lab and AFRL 711HPW/RHWTE for their help in this project. **Funding:** This work was supported by the AIM-MIER project via Infoscitex Corporation (IST) and Air Force Research Laboratory (AFRL) under Air Force contract FA8650-20-D-6207.

Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government. Cleared for public release on 13 Feb 2024. Originator reference number RH-24-125355. Case number AFRL-2024-0832.

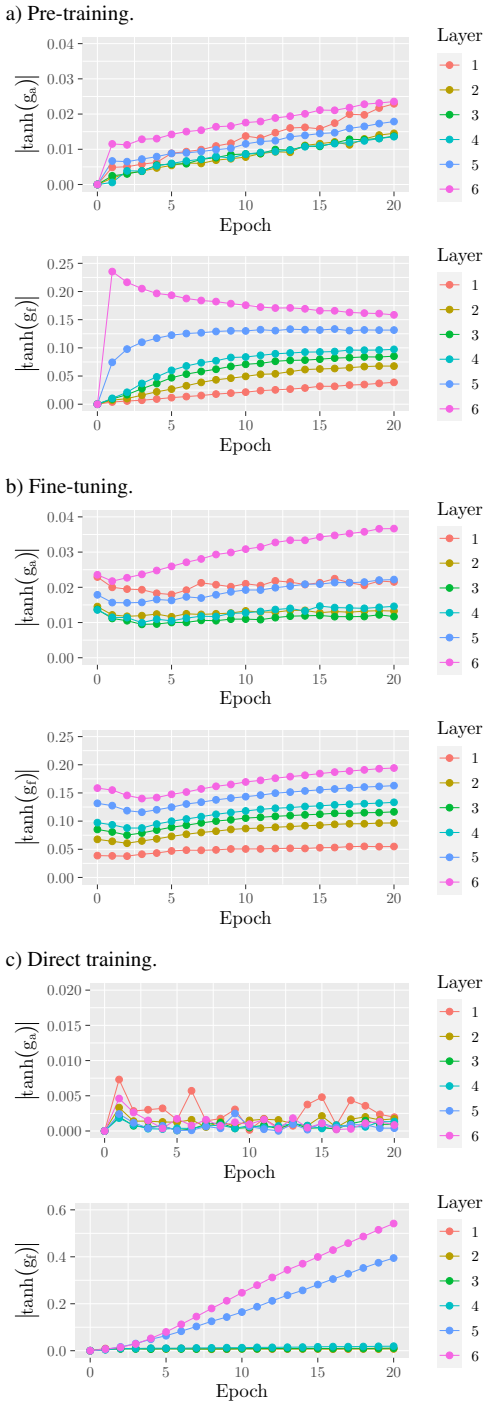

Figure 3: Gating values during a) pre-training over the CR dataset, b) fine-tuning over the Multi30k dataset, and c) directly training on the Multi30k dataset. Layer 1 is the vision-text adapter layer that is closest to the input. Note that some of the gating values overlap in some of the plots.

References

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., and Simonyan, K. (2022). Flamingo: a Visual Language Model for Few-Shot Learning.

Caglayan, O., Kuyu, M., Amac, M. S., Madhyastha, P., Erdem, E., Erdem, A., and Specia, L. (2021). Cross-lingual visual pre-training for multimodal machine translation. In Merlo, P., Tiedemann, J., and Tsarfaty, R., editors, *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.

Caglayan, O., Madhyastha, P., Specia, L., and Barrault, L. (2019). Probing the Need for Visual Context in Multimodal Machine Translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4159–4170, Minneapolis, Minnesota. Association for Computational Linguistics.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*, page 213–229, Berlin, Heidelberg. Springer-Verlag.

Elliott, D., Frank, S., Sima'an, K., and Specia, L. (2016). Multi30K: Multilingual English-German Image Descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74. Association for Computational Linguistics. event-place: Berlin, Germany.

Futeral, M., Schmid, C., Laptev, I., Sagot, B., and Bawden, R. (2023). Tackling ambiguity with images: Improved multimodal machine translation and contrastive evaluation. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5394–5413, Toronto, Canada. Association for Computational Linguistics.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. (2019). Parameter-Efficient Transfer Learning for NLP. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Ive, J., Madhyastha, P., and Specia, L. (2019). Distilling translations with visual awareness. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6525–6538, Florence, Italy. Association for Computational Linguistics.

Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., and Carreira, J. (2021). Perceiver: General Perception with Iterative Attention. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4651–4664. PMLR.

Kocmi, T., Bawden, R., Bojar, O., Dvorkovich, A., Federmann, C., Fishel, M., Gowda, T., Graham, Y., Grundkiewicz, R., Haddow, B., Knowles, R., Koehn, P., Monz, C., Morishita, M., Nagata, M., Nakazawa, T., Novák, M., Popel, M., and Popović, M. (2022). Findings of the 2022 conference on machine translation (WMT22). In Koehn, P., Barrault, L., Bojar, O., Bougares, F., Chatterjee, R., Costa-jussà, M. R., Federmann,

C., Fishel, M., Fraser, A., Freitag, M., Graham, Y., Grundkiewicz, R., Guzman, P., Haddow, B., Huck, M., Jimeno Yepes, A., Kocmi, T., Martins, A., Morishita, M., Monz, C., Nagata, M., Nakazawa, T., Negri, M., Névéol, A., Neves, M., Popel, M., Turchi, M., and Zampieri, M., editors, *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Li, Y., Panda, R., Kim, Y., Chen, C., Feris, R., Cox, D., and Vasconcelos, N. (2022). Valhalla: Visual hallucination for machine translation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5206–5216, Los Alamitos, CA, USA. IEEE Computer Society.

Lin, H., Meng, F., Su, J., Yin, Y., Yang, Z., Ge, Y., Zhou, J., and Luo, J. (2020). Dynamic Context-guided Capsule Network for Multimodal Machine Translation. *Proceedings of the 28th ACM International Conference on Multimedia*. ISBN: 9781450379885 Publisher: ACM.

Ng, N., Yee, K., Baevski, A., Ott, M., Auli, M., and Edunov, S. (2019). Facebook FAIR's WMT19 news translation task submission. In Bojar, O., Chatterjee, R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Yepes, A. J., Koehn, P., Martins, A., Monz, C., Negri, M., Névéol, A., Neves, M., Post, M., Turchi, M., and Verspoor, K., editors, *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. (2016). Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models. *International Journal of Computer Vision*, 123:74 – 93.

Post, M. (2018). A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning Transferable Visual Models From Natural Language Supervision. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Sharma, P., Ding, N., Goodman, S., and Soricut, R. (2018). Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Vijayan, V., Bowen, B., Grigsby, S., Anderson, T., and Gwinnup, J. (2024). The case for evaluating multimodal translation models on text datasets. *arXiv:2403.03014 [cs.CL]*.

Wang, D. and Xiong, D. (2021). Efficient Object-Level Visual Context Modeling for Multimodal Machine Translation: Masking Irrelevant Objects Helps Grounding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(4):2720–2728.

Wu, Z., Kong, L., Bi, W., Li, X., and Kao, B. (2021). Good for misconceived reasons: An empirical revisiting on the need for visual context in multimodal machine translation. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6153–6166, Online. Association for Computational Linguistics.

Yao, S. and Wan, X. (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.

Yin, Y., Meng, F., Su, J., Zhou, C., Yang, Z., Zhou, J., and Luo, J. (2020). A Novel Graph-based Multi-modal Fusion Encoder for Neural Machine Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3035, Online. Association for Computational Linguistics.

Zhang, P., Li, X., Hu, X., Yang, J., Zhang, L., Wang, L., Choi, Y., and Gao, J. (2021). VinVL: Revisiting Visual Representations in Vision-Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5579–5588. _eprint: 2101.00529.

Zhang, Z., Chen, K., Wang, R., Utiyama, M., Sumita, E., Li, Z., and Zhao, H. (2020). Neural Machine Translation with Universal Visual Representation. In *International Conference on Learning Representations*.

A Number of parameters in the GRAM model

The number of parameters in the original text-only Transformer is 269,746,176. While there are also 304,293,888 parameters in the ViT-L/14@336px CLIP vision encoder model that we use, the vision encoder is not used during training since we cache the image encodings to file. We add 68,051,980 parameters via the perceiver resampler and the six vision-text layers, which are that parameters that we optimize over. Thus, the entire model contains 337,798,156 parameters. If we include the vision encoder as well, then the entire model contains 642,092,044 parameters.

B Datasets

Dataset	Only text	With image	Total
CR	1,183,301	5,542,330	7,725,631
M30k	29,000	29,000	58,000

Table 2: Training datasets used in this work. CR is the augmented Conceptual Captions and RAPID2019 datasets described in Section 3.2.1 that we use for pre-training. M30k is the augmented Multi30k dataset used for fine-tuning and is described in Section 3.2.2. "Only text" is the number of examples in the dataset with no associated image. "With image" is the number of examples with one or more associated images. "Total" is the total number of examples in the dataset.

C Optimization details

C.1 Pre-training

We use the same optimization hyper-parameters as FAIR's WMT19 model (Ng et al., 2019) with Fairseq (Ott et al., 2019) as the training and evaluation framework. For pre-training, we use the Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.98$, with a warm-up phase of 4,000 steps where we linearly increase the learning rate from 10^{-7} to 0.0007. Each training batch contains 3,584 source/target tokens and we train for 20 epochs. We use the checkpoint from the last epoch for fine-tuning.

C.2 Training against Multi30k

Fine-tuning. When we perform fine-tuning, we lower the learning rate to 0.0002 and train for 20 epochs. Since the Multi30k dataset is small, we use a warm-up phase of 240 steps where we linearly increase the learning rate from 10^{-7} to 0.0002. We select the checkpoint that performs best against the Multi30k validation set with respect to BLEU4 score.

Direct training. When we directly train, we set the learning rate to 0.0007 and train for 20 epochs using a warm-up phase of 240 steps.

D Model variations

D.1 Where to insert the vision-text adapter layers

For the GRAM model, vision-text cross-attention adapter layers can be added before each of the layers in the Transformer model. Since we modify an encoder-decoder Transformer in order to transform it from an MT model to an MMT model, there are three options for where we add the vision-text layers. One is to insert the vision-text layers before each layer in the Transformer encoder (M_{CR}). Second is to insert the vision-text layers before each layer in the Transformer decoder (DO_{CR}). Third is to insert the vision-text layers before each layer in both the Transformer encoder and decoder (ED_{CR}).

We compare the performance of the three options, the results which are in Table 3. We selected the M_{CR} for fine-tuning since the CoMMuTE score was 0.57 compared to CoMMuTE score of 0.55 for DO_{CR} and 0.52 for ED_{CR}.

D.2 Smaller model variations when directly training against Multi30k

We also explored smaller models when directly training against Multi30k due to the small size of the dataset. For the first smaller model, we set the number of attention heads to 8 and intermediate feed-forward layer size to 2,048 of the vision-text cross-attention layers (S_{M30k_o} and S_{M30k}). For the second smaller model, we set the number of attention heads to 4 and intermediate feed-forward layer size to 1,024 of the vision-text cross-attention layers (T_{M30k_o} and T_{M30k}). As shown in Table 4, we found performance to be similar.

Label	PT	FT	CoMMuTE	Multi30k				newstest	
				2016	2017	coco	2018	2019	2020
			Score	BLEU4					
Multimodal inputs									
M_{CR}	CR		0.57	35.08	39.17	36.79	31.45	35.72	
DO_{CR}	CR		0.55	32.59	41.16	37.54	33.46	36.64	
ED_{CR}	CR		0.52	34.14	39.56	37.45	31.34	35.94	
Text inputs only									
FAIR-WMT19			0.50	32.63	40.66	37.70	33.97	36.45	40.62	36.20
M_{CR}	CR		0.50	31.98	40.22	37.75	32.81	36.41	40.56	35.35
DO_{CR}	CR		0.50	30.01	40.85	37.19	33.36	35.84	38.36	33.79
ED_{CR}	CR		0.50	30.61	40.03	37.80	32.34	36.11	40.18	34.15
Non-matching inputs									
M_{CR}	CR		0.51	30.37	39.01	36.73	32.10	35.35	42.09	35.62
DO_{CR}	CR		0.50	33.07	41.02	37.72	33.54	36.59	42.17	36.20
ED_{CR}	CR		0.50	34.02	39.67	37.44	31.19	35.74	40.84	34.95

Table 3: Performance results of our model under various pre-training and fine-tuning conditions for English to German (en-de) translations. The label FAIR-WMT19 shows our model's performance before our training process, i.e., the original text-only Transformer's performance. M_{CR} is our model pre-trained on the CR dataset. DO_{CR} and ED_{CR} are variations where the vision-text layers are inserted before the decoder layers only (DO_{CR}) and inserted before both the encoder and decoder layers (ED_{CR}), while the M_{CR} model is the variation where the vision-text layers are inserted before the encoder layers only. "Text inputs only" shows the performances of our model when only the source text is given and a zero vector is given as the vision encoding. "Multimodal inputs" shows the performances of our model when both source text and image is used as input. The model is evaluated against the CoMMuTE (Futeral et al., 2023) testing set, the Multi30k (Elliott et al., 2016) sets, and the newstest (Kocmi et al., 2022) testing sets using BLEU4, calculated using SacreBLEU (Post, 2018). Both CoMMuTE score and BLEU4 scores against the CoMMuTE test dataset are shown for completeness; since the CoMMuTE sentences are very short, the BLEU4 score for CoMMuTE should be weighed lightly. PT indicates pre-training and FT indicates fine-tuning. The datasets used for pre-training and fine-tuning are described in Table 2.

E Dataset variations

We explore four variations of our model where we fine-tune against four datasets: M30k$_o$, M30k, M30k$_o$/ncv14, and M30k/ncv14 (Table 6). The results are shown in Table 6.

M30k$_o$ is the original Multi30k dataset. However, we train our model using a concatenation of the Multi30k training set with images and the Multi30k training set without images. This is to account for evaluation artifacts where the model performance when given both text and image input is higher than model performance with only text input, but the result is only due to the model overfitting on training data that only has (source text, target text, image) triplets and no examples of (source text, target text, $\varnothing$) triplets.

M30k is the Multi30k dataset with vision-based masking of the source sentences as done in Section 3.2.1. For each (source text, target text, image), we search for topic phrases (see Section 3.2.1) in the source sentence and replace each instance of the topic phrase with the <unk> token. We also concatenate the original Multi30k dataset with the (source text, target text, image(s)) triplets and the Multi30k dataset with images removed (source text, target text, $\varnothing$) to this.

M30k$_o$/ncv14 and M30k/ncv14 are the concatenation of M30k$_o$ and M30k, respectively, to the news commentary v14 dataset. The news commentary v14, a news translation dataset comprising X sentence pairs, has been used by Ng et al. (2019) in their

Label	PT	FT	CoMMuTE	Multi30k				newstest	
				2016	2017	coco	2018	2019	2020
			Score	BLEU4					
Multimodal inputs									
M_{M30k_o}	M30k_o		0.50	31.99	45.52	42.20	37.51	39.30	
M_{M30k}	M30k		0.50	27.12	45.93	42.76	37.64	38.82	
S_{M30k_o}	M30k_o		0.50	33.61	46.41	42.29	37.83	39.71	
S_{M30k}	M30k		0.50	29.70	46.09	41.61	38.58	38.98	
T_{M30k_o}	M30k_o		0.50	33.06	46.74	42.44	38.06	39.32	
T_{M30k}	M30k		0.50	27.12	45.93	42.76	37.64	38.82	
Text inputs only									
M_{M30k_o}	M30k_o		0.50	31.99	45.52	42.20	37.51	39.30	37.77	28.30
M_{M30k}	M30k		0.50	27.12	45.93	42.76	37.64	38.82	36.09	26.81
S_{M30k_o}	M30k_o		0.50	33.61	46.41	42.29	37.83	39.71	37.75	27.59
S_{M30k}	M30k		0.50	29.70	46.09	41.61	38.58	38.98	36.71	27.89
T_{M30k_o}	M30k_o		0.50	33.06	46.74	42.44	38.06	39.32	37.09	28.12
T_{M30k}	M30k		0.50	29.38	46.21	42.20	38.08	38.88	37.37	28.21
Non-matching inputs									
M_{M30k_o}	M30k_o		0.50	31.99	45.52	42.20	37.51	39.30	37.77	28.30
M_{M30k}	M30k		0.50	27.12	45.93	42.76	37.64	38.82	36.09	26.81
S_{M30k_o}	M30k_o		0.50	33.61	46.41	42.29	37.83	39.71	37.75	27.59
S_{M30k}	M30k		0.50	29.70	46.09	41.61	38.58	38.98	36.71	27.89
T_{M30k_o}	M30k_o		0.50	33.06	46.74	42.44	38.06	39.32	37.09	28.12
T_{M30k}	M30k		0.50	29.38	46.21	42.20	38.08	38.88	37.37	28.21

Table 4: Performance results of our model under various pre-training and fine-tuning conditions for English to German (en-de) translations. The label FAIR-WMT19 shows our model's performance before our training process, i.e., the original text-only Transformer's performance. M_{CR} is our model pre-trained on the CR dataset; $M_{\text{CR,M30k}}$ is our model pre-trained on CR and fine-tuned on Multi30k; M_{M30k} is our model trained on Multi30k without the pre-training step. S_{M30k} and T_{M30k} are smaller variations of the M_{M30k} model. The datasets used for pre-training and fine-tuning are described in Table 2.

fine-tuning step in order to perform well against the newstest testing sets.

Optimization details for the dataset variants. When we perform fine-tuning, we lower the learning rate to 0.0002 and train for 20 epochs. Since the Multi30k dataset is small, for M30k_o and M30k we use a warm-up phase of 240 steps where we linearly increase the learning rate from 10^{-7} to 0.0002. We select the checkpoint that performs best against the Multi30k validation set with respect to BLEU4 score. For M30k_o/ncv14 and M30k/ncv14, we use a warm-up phase of 1200 steps where we linearly increase the learning rate from 10^{-7} to 0.0002. We create a validation set from the concatenation of the WMT19 validation set and the Multi30k validation set and select the checkpoint that performs best against the validation set with respect to BLEU4 score.

E.1 Simultaneously fine-tuning Multi30k and a text-only dataset

Since the pre-training step does degrade performance on the newstest datasets (e.g., 36.2 BLEU4 on newstest2020 for the text-only FAIR-WMT19 model compared to 35.4 BLEU4 for the M_{CR} model), and fine-tuning against Multi30k alone only slightly improves this performance, we explore how to fine-tune our model such that we preserve the performance on the Multi30k test sets and improve the performance on the newstest datasets.

Ng et al. (2019) used the news commentary dataset (Kocmi et al., 2022), a news translation dataset, as the final fine-tuning step in order to

improve performance against the newstest datasets. Similarly, we perform fine-tuning on a concatenation of the Multi30k and news commentary v14 dataset, which resulted in improvements in both the newstest datasets and the Multi30k test sets (e.g., 35.4 BLEU4 on newstest2020 for the M_{CR} model compared to 36.2 BLEU4 for the $M_{\text{CR,M30k/ncv14}}$ model).

E.2 Fine-tuning without vision-based masking of source text

Since most of the captions in Multi30k do not require the image in order to be correctly translated due to the captions being unambiguous (Futeral et al., 2023), MMT models tend to ignore visual information during the training process (Caglayan et al., 2019; Wu et al., 2021). We are able to quantitatively see this when directly training against the original Multi30k dataset (for M_{M30k_o}, the CoMMuTE score is 0.5).

So we ask ourselves how we may preserve CoMMuTE performance along with newstest and Multi30k test performances. Since vision-based masking of source sentences was used to improve performance during the pre-training stage, we explore whether it can improve performance during the fine-tuning stage as well.

Thus, we create the M30k and the M30k/ncv14 datasets as described above. The M30k contains masked source sentences from the Multi30k dataset and the M30k/ncv14 dataset is a concatenation of the M30k and the text-only news commentary v14 datasets. We see that fine-tuning using these datasets preserve the CoMMuTE score much better than when not using informed masking (Table 6) while only slightly decreasing BLEU4 scores.

Label	PT	FT	CoMMuTE	Multi30k				newstest	
				2016	2017	coco	2018	2019	2020
			Score	BLEU4					
Multimodal inputs									
M_{CR}	CR		0.57	35.08	39.17	36.79	31.45	35.72	
$M_{\text{CR,M30k}_o}$	CR	M30k$_o$	0.58	33.03	47.11	43.75	39.48	40.94	
$M_{\text{CR,M30k}}$	CR	M30k	0.61	35.03	46.50	43.57	39.10	40.40	
$M_{\text{CR,M30k}_o\text{/ncv14}}$	CR	M30k$_o$/ncv14	0.58	33.99	47.38	42.95	39.83	40.92	
$M_{\text{CR,M30k/ncv14}}$	CR	M30k/ncv14	0.63	34.88	46.57	43.58	39.78	41.03	
M_{M30k_o}	M30k$_o$		0.50	31.99	45.52	42.20	37.51	39.30	
M_{M30k}	M30k		0.50	27.12	45.93	42.76	37.64	38.82	
Text inputs only									
FAIR-WMT19			0.50	32.63	40.66	37.70	33.97	36.45	40.62	36.20
M_{CR}	CR		0.50	31.98	40.22	37.75	32.81	36.41	40.56	35.35
$M_{\text{CR,M30k}_o}$	CR	M30k$_o$	0.50	31.25	47.10	43.08	38.48	40.82	42.64	36.00
$M_{\text{CR,M30k}}$	CR	M30k	0.50	32.11	46.43	42.88	37.88	40.35	42.66	36.22
$M_{\text{CR,M30k}_o\text{/ncv14}}$	CR	M30k$_o$/ncv14	0.50	31.17	47.40	43.30	38.86	40.70	41.80	36.44
$M_{\text{CR,M30k/ncv14}}$	CR	M30k/ncv14	0.50	32.95	46.65	43.06	38.95	40.73	41.86	36.46
M_{M30k_o}	M30k$_o$		0.50	31.99	45.52	42.20	37.51	39.30	37.77	28.30
M_{M30k}	M30k		0.50	27.12	45.93	42.76	37.64	38.82	36.09	26.81
Non-matching inputs									
M_{CR}	CR		0.51	30.37	39.01	36.73	32.10	35.35	42.09	35.62
$M_{\text{CR,M30k}_o}$	CR	M30k$_o$	0.52	32.17	47.08	42.97	38.55	41.12	42.31	36.12
$M_{\text{CR,M30k}}$	CR	M30k	0.51	31.22	46.56	43.19	37.94	40.75	42.04	36.18
$M_{\text{CR,M30k}_o\text{/ncv14}}$	CR	M30k$_o$/ncv14	0.50	29.39	47.24	43.44	39.48	41.11	41.82	36.52
$M_{\text{CR,M30k/ncv14}}$	CR	M30k/ncv14	0.51	31.69	46.37	43.06	38.90	40.72	41.78	36.27
M_{M30k_o}	M30k$_o$		0.50	31.99	45.52	42.20	37.51	39.30	37.77	28.30
M_{M30k}	M30k		0.50	27.12	45.93	42.76	37.64	38.82	36.09	26.81

Table 6: Performance results of our model under various pre-training and fine-tuning conditions for English to German (en-de) translations. The label FAIR-WMT19 shows our model's performance before our training process, i.e., the original text-only Transformer's performance. M_{CR} is our model pre-trained on the CR dataset; $M_{\text{CR,M30k}}$ is our model pre-trained on CR and fine-tuned on Multi30k; M_{M30k} is our model trained on Multi30k without the pre-training step. The datasets used for pre-training and fine-tuning are described in Table 2.