This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

You Need Multiple Exiting: Dynamic Early Exiting for Accelerating Unified Vision Language Model

Shengkun Tang[1], Yaqing Wang[2], Zhenglun Kong[6], Tianchi Zhang[4], Yao Li[5], Caiwen Ding[3]
Yanzhi Wang[6], Yi Liang[2], Dongkuan Xu[1]
[1]North Carolina State University, Raleigh, USA [2]Google Research, New York, USA
[3]University of Connecticut, Mansfield, USA [4]University of Michigan, Ann Arbor, USA
[5]The University of North Carolina at Chapel Hill, Chapel Hill, USA [6]Northeastern University, Boston, USA
shengkuntangwork@gmail.com, {yaqingwang, yiliang}@google.com, yaoli@email.unc.edu
{kong.zhe, yanz.wang}@northeastern.edu, tonyztc@umich.edu, caiwen.ding@uconn.edu, dxu27@ncsu.edu

Abstract

*Large-scale Transformer models bring significant improvements for various downstream vision language tasks with a unified architecture. The performance improvements come with increasing model size, resulting in slow inference speed and increased cost for severing. While some certain predictions benefit from the full computation of the large-scale model, not all of inputs need the same amount of computation to conduct, potentially leading to computation resource waste. To handle this challenge, early exiting is proposed to adaptively allocate computational power in term of input complexity to improve inference efficiency. The existing early exiting strategies usually adopt output confidence based on intermediate layers as a proxy of input complexity to incur the decision of skipping following layers. However, such strategies cannot be applied to encoder in the widely-used unified architecture with both encoder and decoder due to difficulty of output confidence estimation in the encoder layers. It is suboptimal in term of saving computation power to ignore the early exiting in encoder component. To address this issue, we propose a novel early exiting strategy for unified vision language models, which allows to dynamically skip the layers in encoder and decoder simultaneously in term of input layer-wise similarities with multiple times of early exiting, namely **MuE**. By decomposing the image and text modalities in the encoder, MuE is flexible and can skip different layers in term of modalities, advancing the inference efficiency while minimizing performance drop. Experiments on the SNLI-VE and MS COCO datasets show that the proposed approach MuE can reduce expected inference time by up to 50% and 40% while maintaining 99% and 96% performance respectively.*

1. Introduction

Recent advances in multi-modal Transformer-based large-scale models [25, 28, 48, 57] bring improvements across various vision language tasks. Among the Transformer-based models, the unified sequence-to-sequence architecture [38, 46] has attracted much attention due to its potential to become a universal computation engine to diverse tasks. Although large-scale models have achieved unattainable performance, their expensive computational cost usually hinders their applications in real-time scenarios.

While the scaling effect suggests that the performance of the model benefits from its increased size, not every input requires the same amount of computation to achieve similar performance. Such an observation is particularly valid in visual language tasks, where inputs from different modalities may require different amounts of computation. Early exiting is a popular method to reduce computational costs by adaptively skipping layers on top of the input while preserving the general knowledge of the large-scale model. Existing studies aim to deal with early exiting decisions for encoder-only models [51, 52] or decoder components in encoder-decoder architectures [8], but cannot induce early exiting decisions for both components at the same time. Considering that single-component strategies may be suboptimal in terms of saving computation cost, in this paper, we investigate how to perform early exiting for both encoder and decoder components in a sequence-to-sequence architecture to elucidate a new way to further improve inference efficiency.

Given the varied complexity of the inputs, it is natural to consider skipping some layers of the encoder as well as the decoder. Current decision mechanisms use classifiers to predict the output confidence of intermediate representations and stop computation if the confidence reaches

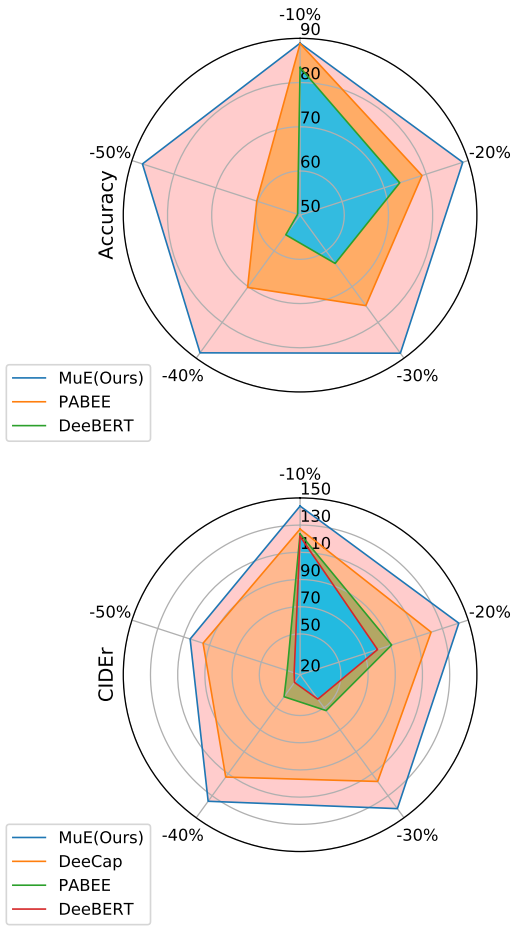

Figure 1. The performance of different early exiting methods on SNLI-VE [49] and MS COCO [1] with certain expected inference time reduction rates.

predefined threshold. However, extending this to unified sequence-to-sequence model is non-trivial due to two main challenges: (1) there are dependencies in making decisions for exiting decisions in the encoder and decoder, and (2) it is difficult to apply confidence classifiers to skip the encoder layer before going through the decoder layer for task output. To address the above challenges and to enable early exiting of encoder and decoder in sequence-to-sequence framework, we propose a novel early exiting strategies based on layer-wise input similarity, which is different from existing works based on task confidence [42]. More specifically, the model is encouraged to skip following layers in both encoder and decoder when the layer-wise similarity reaches a certain threshold.

This method is inspired by the saturation observation [11] which shows that the hidden-state of each Transformer layer arrives at a saturation status as going into deep layers. For the vision-language tasks, we find that the similar observation regarding saturation is also valid as shown in Figure 3. This observation lands the foundation that we could make the exiting decision based on the intermediate layer-wise input similarities without going through the following layers. Besides, since the computation needed for input in different modalities usually varies, we propose a modality decomposition mechanism, which could further enable early fusion large-scale multi-modal models to break tie between modalities and enjoy the flexible exiting decision over modalities. To encourage the early exiting behavior with a minimal performance loss, we design a layer-wise task loss, which enforce the each layer to output informative features for final task. Figure 1 shows the results on SNLI-VE dataset [49] and MS COCO [1] in term of expected time reduction rate and task performance. We compare MuE with several State-of-the-art early existing methods and observe that MuE is able to largely reduce inference time with a minimal performance drop compared to other SoTA methods. Our main contributions are summarized as follows:

- To the best of our knowledge, this is a pioneering work to extend early exiting choices to both encoder and decoder of sequence-to-sequence architecture. To this end, we propose a novel early exiting strategy based on layer-wise input similarity with the valid assumption on saturation states in vision language models.
- Given the different characteristics of the modalities and tasks, we decompose the modalities encoder for early-fusion pre-trained models, bringing additional improvements in terms of inference efficiency.
- We introduce layer-wise task loss, linking each layer in the decoder to the final task, effectively helping to maintain task performance when a significant time reduction is required.
- Extensive experiments show that our method can largely reduce the inference time of vision language models by up to 50% with minimal performance drop.

2. Related Work

Vision Language. Vision language learning recently attracts lots of attention [2,5,10,18,22,25,29,34,37,43,47,53,54,57–59], where Transformer-based model [21,33] brings significant improvements to diverse downstream vision language tasks. Recent efforts [25,28,48,55–57] leverage the sequence-to-sequence architecture to unify diverse tasks in a generation manner to build the universal computation engine. With the unified framework, the recent study [46] shows promising results on several vision language benchmark datasets via introducing instructions to handle diverse tasks. Even though the large-scale vision language models achieve unattainable performance, the expensive inference costs still hinder their application in real-world scenarios.

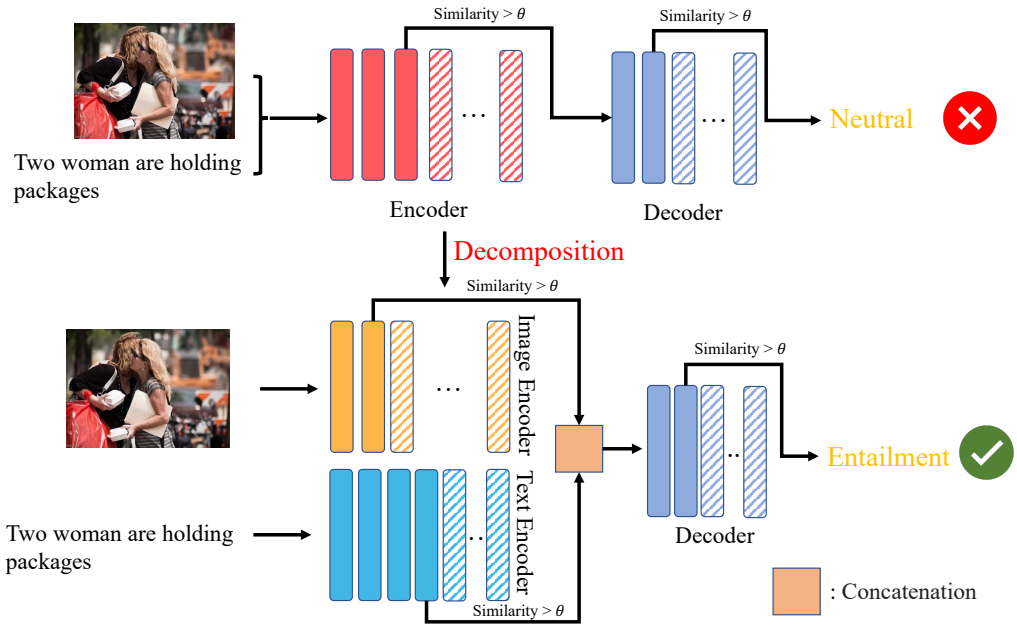

Figure 2. The overview of our proposed MuE.

Early Exiting Strategy. How to improve inference efficiency of large-scale models attracts lots of attention in recent years with existing efforts including knowledge distillation [7, 13], quantization [32], weight pruning [9] and others [17, 23, 26, 40]. In this paper, we focus on early exiting [6, 14, 15, 19, 23, 31, 35, 36, 44, 52], which aims to dynamically allocate computation resource per example, with the goal of improving the inference speed while minimizing the performance drop. DeeBERT [51] and FastBERT [30] explore early exiting strategy with BERT [4] model and several efforts are on varied natural language processing tasks including natural language understanding [20, 27, 60], sequence labeling [24] and document ranking [50]. For the vision language learning, DeeCap [8] explores early exiting in image captioning tasks with a shallow imitation network to help maintain the performance. However, DeeCap is dedicated to decoder layers only and cannot be easily extended to encoders, thereby limiting the room of further advancing the inference efficiency. To handle this challenge, we propose a novel early exiting strategy, which introduces this computation resource allocation process in both encoder and decoder, while minimizing the performance drop.

3. Method

The objective of early exiting is to dynamically allocate the computation power in term of input complexity to improve the inference efficiency while maintaining the performance. Considering a unified visual language model consisting of encoder and decoder, we propose an early exiting method which is able to skip layers in both encoder and decoder components. The input to the visual language model includes images and text, while the output varies with the need of downstream tasks, such as the textual sentence for image captioning and class prediction for visual entailment.

The overview of the proposed approach is given in Figure 2. To allow flexibility in the decision to skip different modalities, we discuss how to decompose the modalities in Sec. 3.1. Then, we describe how to use layer-wise similarity to guide early exiting of encoder and decoder layers in Sec. 3.2. In Sec. 3.3, we introduce layer-wise task loss, which can be effective in helping to maintain performance when a significant reduction in inference cost is required.

3.1. Modality Decomposition

We introduce how to decompose the early-fusion encoder to modality-specific encoder to allow processing image and text independently during fine-tuning and inference stages. We first describe how the image and text are processed in the early-fusion encoder of unified vision language architecture. Following the recent work [46], we represent text and image in tokens. Let the input token embedding representations of image be $\mathbf{I}_0$ and text be $\mathbf{T}_0$. Before feeding into transformer layers, the image tokens are processed by ResNet [12]. As for positional information, we decouple the position correlation from image embeddings and text embeddings [16]. Moreover, we use 2D relative position bias [3, 48] for image and 1D relative position bias for text [39], respectively. The full input sequence includes

the concatenation of token representations from image and text tokens as $[\mathbf{I}_0; \mathbf{T}_0]$. The Transformer encoder has n layers (denoted E_i for layer i) and we refer the readers to [45] for the details of the Transformer layer. Thus, the process can be formulated as:

$$[\mathbf{I}_i; \mathbf{T}_i] = E_i([\mathbf{I}_{i-1}; \mathbf{T}_{i-1}]). \tag{1}$$

We denote a stack of layers from layer i to layer j in encoder as $E_{i:j}$. The image $\mathbf{I}_n$ and text input $\mathbf{T}_n$ through the encoder without decomposition strategy can be written as:

$$[\mathbf{I}_n; \mathbf{T}_n] = E_{1:n}([\mathbf{I}_0; \mathbf{T}_0]). \tag{2}$$

To decompose the encoder into modality-specific encoder, we duplicate the encoder to handle the input, where the image tokens and text tokens are fed into two encoder respectively. The parameters of the two encoder are tied without introducing extra parameters. The output representations of the decomposed encoder, $\mathbf{I}_n$ and $\mathbf{T}_n$ can be expressed as:

$$[\mathbf{I}_n; \mathbf{T}_n] = [E_{1:n}(\mathbf{I}_0); E_{1:n}(\mathbf{T}_0)]. \tag{3}$$

Encoder decomposition enables the proposed model to process the image and text independently, which in turn allows us to apply early exiting for different modalities by necessity and pick up essential information of each modality for different tasks during inference. When the image encoder exits at layer p and the text encoder exits at layer q, The output of the encoder part can be written as:

$$[\mathbf{I}_p; \mathbf{T}_q] = [E_{1:p}(\mathbf{I}_0); E_{1:q}(\mathbf{T}_0)]. \tag{4}$$

After decomposition, the image and text tokens are concatenated and fed into the decoder, denoted as $C = [\mathbf{I_p}; \mathbf{T_q}]$. The decoder contains a self-attention module and a cross-attention module in each layer. We denote the input of self-attention module as $Td_{i,s}, i = 0$. s refers to the step in decoding. In the layer i, The output representation of self-attention module and C will be sent into cross-attention module, which is formulated as following:

$$Td_{i,s} = \text{CrossAttn}(\text{SelfAttn}(Td_{i-1,s}), C), \tag{5}$$
$$Td_{0,s} = Td_{n,s-1}. \tag{6}$$

The final output is decoded via a unified vocabulary following [46] to obtain the textual output.

3.2. Early Exiting Based on Layer-wise Similarities

Existing early exiting methods [8, 42, 51] usually add a classifier for each layer to simulate output for input complexity estimation. During inference, the output representations of each layer are fed into the classifier and the confidence or entropy are utilized as a proxy for early exiting decision. However, the classifier cannot be applied to the encoder part in encoder-decoder framework since the hidden-state representation of encoder is not directly related to the final prediction considering that the hidden-state representation will still go through the following decoder. According to [11], the hidden-state representation of each layer in Transformer will reach a saturation status in language model, which indicates that the hidden-state representation change decreases as going through the latter layers. Therefore, the latter layers can be skipped safely without significant performance drop when such saturation status is reached.

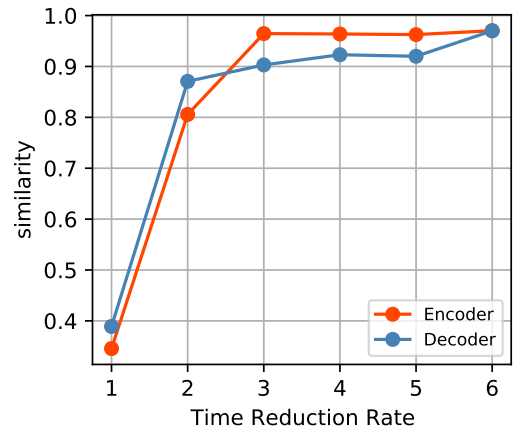

Figure 3. The saturation status in vision language model. The similarities between hidden-states reach the top at an early stage and stay stable in the following layers.

We study whether the saturation state exists in multimodal sequence-to-sequence models, as shown in Figure 3, where the Cosine Similarity of layers is evaluated on the test set of SNLI-VE dataset [49]. As observed in Figure 3, the shallow layers of encoder and decoder show the low similarity level, indicating that the token representation changes dramatically in shallow layers. The similarities reach peak and remain flat in the following layers, confirming the existence of saturation. Based on this observation, we leverage the Cosine Similarity between layers as a proxy to estimate the saturation level:

$$\text{CosSim}(T_i, T_j) = \frac{T_i \cdot T_j}{\|T_i\| \cdot \|T_j\|}. \tag{7}$$

We denote each layer in encoder and decoder as E_i and $D_i (i < n)$. Image token, text token and decoder token representation are $I_i, T_i, Td_{i,s}$, respectively. The similarity can be formulated as:

$$\text{ImgSim}_i = \text{CosSim}(E_i(\mathbf{I}_i), E_{i-1}(\mathbf{I}_{i-1})), \tag{8}$$
$$\text{TxtSim}_i = \text{CosSim}(E_i(\mathbf{T}_i), E_{i-1}(\mathbf{T}_{i-1})), \tag{9}$$
$$\text{DecSim}_{i,s} = \text{CosSim}(D_i(Td_{i,s}), D_{i-1}(Td_{i-1,s})). \tag{10}$$

Note that we compute the similarity between output representation of first layer and the input token representation

Algorithm 1 Inference

Input image I, text tokens T, decoder input $T_{d,s}$, exiting threshold θ

$I = ResNet(I); T = Embed(T)$
$img_states = [I]; txt_states = [T]$
# For text encoder.
for $i \leftarrow 1$ to N **do**
  $T = Enc_i(T)$
  if $similarity(txt_states[-1], T) > \theta$ **then**
    break
  $txt_states.append[T]$
# For image encoder.
for $i \leftarrow 1$ to N **do**
  $I = Enc_i(I)$
  if $similarity(img_states[-1], I) > \theta$ **then**
    break
  $img_states.append[I]$
For Decoder.
$Enc_out = Concat[I, T]$; $Dec_states = [T_{d,0}]$
for $s \leftarrow 1$ to S **do**
  for $i \leftarrow 1$ to N **do**
    $T_{d,s} = Dec_i(T_{d,s}; \text{Enc_out})$
    if $similarity(Dec_states[-1], T_{d,s}) > \theta$ **then**
      break
    $Dec_states.append[T_{d,s}]$
Return: $T_{d,s}$

when $i = 1$. In that case, we can evaluate the saturation state of first layer. Once the similarity meets the predefined threshold, the computation will terminate and skip the following layers. During the inference stage in generation tasks such as image captioning, we generate new words in a auto-regressive manner and the proposed early exiting strategy apply to each step of generation. The detailed pseudocodes are shown in Algorithm 1.

During the inference of generation tasks, the errors of each step will accumulate, which degrades the final generation results severely. We reduce errors at the beginning of generations with slight computation increase since the token numbers at early generation stage are relatively small. Therefore, we follow [41] to utilize decay threshold Θ which can be formulated as follows:

$$\Theta(t) = \beta\theta + (1 - \beta)e^{-\tau t/N}, \tag{11}$$

where θ is a pre-define static threshold, t is the timestep of generation, N is the total number of steps, β and τ are all hyperparameters.

3.3. Layer-wise Task Loss

The main assumption of early exiting method is that the intermediate representations of some easy samples in test sets have enough information for final predictions [44, 51]. However, as we show in Sec. 4.2.2, the intermediate hidden states fail to predict the final results in decoding stage of sequence-to-sequence framework. Correspondingly, a significant performance drop is observed with intermediate representation of fine-tuned model. To address this issue, we propose to update every decoder layer of our model simultaneously with task loss to encourage the early exiting behavior and achieve the goal of maintaining the performance.

The existing early exiting methods usually adopt two stage of training, where first one is to fine-tune the model and the following one is to develop an exiting classifier with the frozen fine-tuned checkpoint from the first step. Different from the existing early exiting works, we introduce layer-wise task loss to encourage early exiting behavior during fine-tuning. More specifically, we add loss to each decoder layer and update them via optimizing the task loss. We observe that such a step could effectively improve the efficiency of early exiting and help maintain task performance. The loss function is written as:

$$\mathcal{L} = \frac{1}{N}\sum_{i}^{N} \mathcal{L}_{CE}, \tag{12}$$

$$\mathcal{L}_{CE} = -\sum_{i=1}^{|y|} \log P_\theta(y_i | y_{<i}, \mathbf{I}, \mathbf{T}), \tag{13}$$

where N is the number of decoder layers, $\mathbf{I}$, $\mathbf{T}$ and y denote image tokens, text tokens and output respectively. With this new training objective, the model can be optimized to balance severing final task from intermediate layer and final layer. Therefore, the prediction based on the intermediate is more likely to be informative, reducing the performance gap between early exiting and full model.

4. Experiments

4.1. Experimental Setups

Dataset and Evaluation Metric. SNLI-VE dataset [49] and MS COCO [1] are used to evaluate our methods for visual entailment and image captioning. SNLI-VE dataset provides a pair of image and text and requires the model to determine if the given image and text description are correlated. We report accuracy and expected time reduction rate on dev and test datasets. MS COCO is used for evaluating the image captioning which requires models to generate an appropriate and fluent caption for a given image. The input of our model in image captioning is a image and a piece of instruction text. We report BLEU-4, METEOR, CIDEr, SPICE scores and expected time reduction rate on the Karpathy test split in main results. The full evaluation metrics can be found in supplementary details. As the measurement of runtime might not be stable even in the same

Models	SNLI-VE			Image Captioning				
	Dev	Test	Time	BLEU-4	METEOR	CIDEr	SPICE	Time
OFA$_{Base}$	89.3	89.2	1	42.8	31.7	146.7	25.8	1
OFA$_{Tiny}$	85.3	85.2	-33%	38.1	29.2	128.7	23.1	-33%
DeeBERT	78.9	78.8	-15%	30.1	26.3	102.1	20.5	-15.5%
PABEE	85.3	85.2	-15.3%	31.4	26.8	105.8	21	-16.3%
DeeCap	-	-	-	38.7	29.1	129	22.5	-38%
Ours	**88.7**	**88.5**	**-50%**	**41.6**	**30.6**	**137**	**24.4**	**-40.2%**

Table 1. The performance and expected time reduction rate comparison between our method and previous methods. Our method reduces more computation comparing with other methods while preserves well performance. Time: expected time reduction rate

environment, we propose a new metric to evaluate the efficiency in Seq2Seq architecture, called expected time reduction rate which can be defined as:

$$1 - \frac{\frac{n_E}{N_E} + \frac{\sum_{i=1}^{N_w} w_i \times n_D}{\sum_{i=1}^{N_w} w_i \times N_D}}{2}, \tag{14}$$

where N_E and N_D are the number of encoder and decoder layers in the model, n_E and n_D are the number of layers used in encoder and decoder during inference and w_i is the number of words that exit at n_D of decoder. The new metric is able to reflect the overall computation reduction ratio of Seq2Seq models.

Baselines. To empirically evaluate the efficiency gains enabled by our proposed measurements, we compare with the original **OFA**$_{Base}$ [46] model which includes 6 encoder layers and 6 decoder layers. **OFA**$_{Tiny}$ is the tiny version of OFA which only contains 4 encoder and decoder layers. Since the experiments of **DeeBERT** [51] and **PAEBB** [60] were originally conducted on BERT [4], we implement it in decoder part on OFA architecture. We only compare our model with **DeeCap** [8] in image captioning.

Implementation Details. Our model is based on a unified vision language model called OFA [46]. We utilize the Base model and released pre-trained weight to fine-tune on downstream tasks. There are 6 encoder layers and 6 decoder layers in the Base model. The hidden state size is 768. More hyperparameter settings can be found in supplementary details. All experiments are implemented by PyTorch and conducted on 4 RTX 6000 GPUs.

4.2. Main Results

4.2.1 Evaluation on Visual Entailment

We evaluate our methods on Visual Entailment, which is a classification task. Qualitative comparisons are shown in Table 1. Our method has saved more computation and achieved best performance than any other method in the table. More concretely, our method can reduce nearly 50% computation while only 0.7 points accuracy drop on the test set of SNLI-VE dataset. In comparison, OFA$_{Tiny}$ model can save around 33% computation with a significant performance drop, from 89.2 to 85.2. This is mainly because the low capacity of the tiny model limits the upper bound of performance. At the same time, DeeBERT is able to only reduce around 15% of overall computation since DeeBERT cannot skip layers in encoder part, which requires large computation. Moreover, DeeBERT caused severe damage to accuracy, from 89.2 to 78.8. The reason behind this circumstance is that the shallow features are not sufficient for final prediction, as demonstrated in [8]. Our proposed training objective can overcome the performance drop while no more computation is introduced. PABEE is better than DeeBERT in terms of accuracy with similar expected time reduction rate but still lower than our method. Overall, compared with previous early exiting methods, our model accelerates vision language model most with a slight accuracy reduction in visual entailment.

4.2.2 Evaluation on Image Captioning

We discuss the efficiency and effectiveness of our method in image captioning task. Experiment results are shown in Table 1. Compared with other early exiting methods and tiny models, our model achieves the highest computation reduction, up to 40.2%. Moreover, the performance of our method still ranks top among other methods on BLEU-4, METEOR, CIDEr and SPICE. DeeBERT causes the highest performance drop, showing the pure classifier is unable to identify the suitable time to exit. With performance recovery strategies, DeeCap can obtain better performance than DeeBERT and PABEE with more computation reduction. Our model appends cross-entropy loss on each layer of decoder which optimizes every decoder layer simultaneously. This training objective minimizes the error of a single layer and reduces the overall error accumulation in autoregressive decoding stages. MuE achieves higher performance on every metric comparing with DeeCap, which proves the effectiveness of our training objectives on performance recovery.

Task	Image Layer	Text Layer	BLEU-4	METEOR	CIDERr	SPICE
Image Captioning	6.0	6.0	42.4	31.2	143.9	25.1
	3.1	2.0	32.8	27.4	112.1	20.8
	3.1	6.0	33.1	27.4	112.2	20.7
	6.0	2.0	**42.0**	**31.2**	**143.6**	**25.1**

Task	Image Layer	Text Layer	Dev	Test
Visual Entailment	6	6	88.6	88.7
	2.03	2.9	76.1	75.6
	6	2.9	79.1	79.5
	2.03	6	**88.4**	**88.6**

Table 2. Results of our proposed decomposition strategy. The results show that for visual entailment, more image encoder layers can be safely removed and exited earlier while the text encoder layers for image captioning.

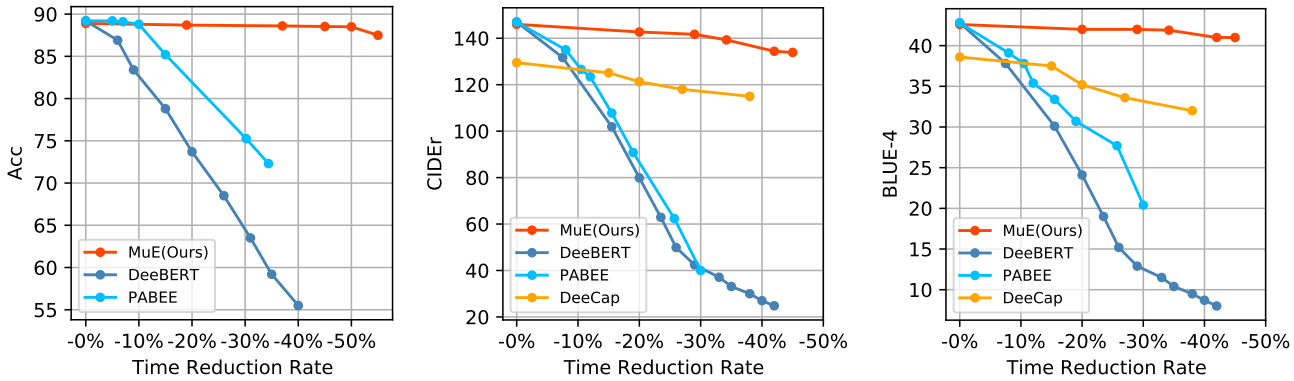

Figure 4. Accuracy and expected time reduction rate trade-off comparison between our proposed MuE and other methods in visual entailment and image captioning.

4.2.3 Effectiveness of Decomposition

We analyze our proposed modality decomposition strategy. In our experiments, we change the early exiting threshold to control the number of encoder layers that are utilized to process the tokens. We report the performance with different combinations of text and image encoder layers in visual entailment and image captioning. The main results are shown in Table 2. The last row is the model with full encoder layers for image and text. The second and third rows skip text and image separately. The first row skips both layers simultaneously. From the table, the accuracy drops dramatically when more text information is lost in visual entailment. The results show that for visual entailment, the understanding of text information is more important than image information. Conversely, in image captioning, all metrics decrease in varied degree if the image layers are skipped more. This demonstrates that the understanding of image information is more essential to image captioning. If more text layers and image layers are skipped at the same time, the performance is dropped in both tasks. Therefore, we should remove image and text layers selectively. Through decomposing image and text tokens in encoder, our model is able to exit earlier if the modality information is unnecessary while keeping more layers when the information is indispensable. For example, the third row of visual entailment table shows that only 2.03 encoder layers that process image tokens is sufficient to obtain 99.8% accuracy. This proves the effectiveness of our proposed decomposition strategy in the encoder.

4.2.4 Accuracy and Computation Trade-off

We analyze the trade-off between the accuracy and computation reduction of our model. According to Table 3, there are several experimental observations. First, the performance on visual entailment and image captioning drops with the increase of computation reduction. At the beginning, the large computation decrease doesn't do harm to performance. With the increase of skipped layers, little reduction in computation brings large amount of performance drop in some metrics such as CIDEr. This makes sense since if we force some mid-difficult samples to exit too early, they will be mistaken by output layers. More-

Models	SNLI-VE			Image Captioning				
	Dev	Test	Time	BLEU-4	METEOR	CIDEr	SPICE	Time
OFA_{Base}	89.3	89.2	1	42.8	31.7	146.7	25.8	1
MuE	88.8	88.7	-19%	42	31.3	142.7	25.1	-20.0%
	88.7	88.6	-37%	42	31.1	141.7	25.1	-28.9%
	88.6	88.5	-45%	41.6	30.7	139.3	24.4	-34.2%
	87.7	87.5	-55%	41	30	133.9	23.6	-45%

Table 3. Trade-off of our proposed MuE.

Models	Test Acc.	Time	CIDEr	Time
MuE Full	88.5	-50%	137.0	-40.2%
MuE w/o Decom	75.0	-35%	117.5	-21%
MuE w/o Task loss	87.0	-47%	117.9	-25%

Table 4. Ablation study in visual entailment and image captioning. The results of our model without decomposition strategy and training objective. Time: time reduction rate

over, we draw the trade-off curves of our methods and other SoTA methods in Figure 1 and Figure 4. As shown in the figure, we can notice that comparing the trade-off curve of other methods, the curve of our method is generally located at the upper right of other curves. It means that with the same computation reduction rate, our method always obtains higher scores. Even at the highest computation drop point, our method is much better than the performance of other methods such as DeeBERT and PABEE at the lowest computation reduction point. Interestingly, at the start point without computation reduction, the performance of our method is slightly lower than DeeBERT and PABEE. This is because we train our decomposition strategy with original fine-tuning and it brings slight degradation on overall performance.

5. Ablation Study

To prove whether our proposed methods are solid and how much these methods contribute to our model separately, we evaluated the model without decomposition strategy and training objective, respectively. The main results are shown in Table 4. The first row is the full model with both decomposition and training strategies. We removed the decomposition strategy in the experiment at the second row. Therefore, in the encoder part, the image and text tokens early exit at the same layer while the decoder layer exiting strategy stays unchanged. In the experiment at the last row, we split training objectives from our full model to show the performance recovery brought by layer-wise task loss. All experiment results are the best trade-off between the scores and expected time reduction rate in visual entailment and image captioning.

As shown in the tables, the model without decomposition strategy gains the lowest test accuracy, CIDEr scores and expected time reduction rate. This is largely due to the information loss of key modality which is the text for visual entailment and image for image captioning. Coarse understanding in the description of the key modality is not enough to make a precise final prediction. Therefore, it is important to keep more layers for key modalities while remove more layers for other layers to improve efficiency. As for training objective, in visual entailment, removing training objective doesn't do harm to the final performance. However, the absence of our proposed layer-wise task loss causes a large drop in both performance and expected time reduction rate in image captioning. This is caused by the accumulation of errors in captioning while there is no accumulation in visual entailment with only one timestep in the decoding stage. Our layer-wise task loss is able to reduce error at every timestep of decoding which is beneficial to final results. The improvement demonstrates the strong ability of our proposed training objective to recover performance.

6. Conclusion

In this paper, we point out that different modality information is required for different tasks in unified vision language models. Moreover, previous early exiting methods cannot only be applied to the encoder part in Seq2Seq frameworks. To meet these features, we propose MuE for accelerating unified vision language models during the inference stage. We decompose the unified encoder into text and image encoders by splitting the input tokens of image text during fine-tuning stage. Similarity between layers is utilized as a metric to determine the exiting time which enables early exiting in both the encoder and decoder part in Seq2Seq architecture. In addition, in order to recover the performance drop caused by early exiting, we propose to train every decoder layer with the same training loss which helps to optimize every layer to the best performance. Experiments demonstrate that our proposed MuE reduces most computation with minimal performance drop in both classification and generation tasks, suggesting the effectiveness of our methods.

References

[1] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 2, 5

[2] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer, 2020. 2

[3] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34:3965–3977, 2021. 3

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3, 6

[5] Zi-Yi Dou, Yichong Xu, Zhe Gan, Jianfeng Wang, Shuohang Wang, Lijuan Wang, Chenguang Zhu, Pengchuan Zhang, Lu Yuan, Nanyun Peng, et al. An empirical study of training end-to-end vision-and-language transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18166–18176, 2022. 2

[6] Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. Depth-adaptive transformer. *arXiv preprint arXiv:1910.10073*, 2019. 3

[7] Zhiyuan Fang, Jianfeng Wang, Xiaowei Hu, Lijuan Wang, Yezhou Yang, and Zicheng Liu. Compressing visual-linguistic model via knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1428–1438, 2021. 3

[8] Zhengcong Fei, Xu Yan, Shuhui Wang, and Qi Tian. Deecap: Dynamic early exiting for efficient image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12216–12226, 2022. 1, 3, 4, 6

[9] Zhe Gan, Yen-Chun Chen, Linjie Li, Tianlong Chen, Yu Cheng, Shuohang Wang, Jingjing Liu, Lijuan Wang, and Zicheng Liu. Playing lottery tickets with vision and language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 652–660, 2022. 3

[10] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. *Advances in Neural Information Processing Systems*, 33:6616–6628, 2020. 2

[11] Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *arXiv preprint arXiv:2203.14680*, 2022. 2, 4

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[13] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 3

[14] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens Van Der Maaten, and Kilian Q Weinberger. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*, 2017. 3

[15] Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. Shallow-deep networks: Understanding and mitigating network overthinking. In *International conference on machine learning*, pages 3301–3310. PMLR, 2019. 3

[16] Guolin Ke, Di He, and Tie-Yan Liu. Rethinking positional encoding in language pre-training. *arXiv preprint arXiv:2006.15595*, 2020. 3

[17] Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Bin Ren, Minghai Qin, Hao Tang, and Yanzhi Wang. Spvit: Enabling faster vision transformers via soft token pruning. *arXiv preprint arXiv:2112.13890*, 2021. 3

[18] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11336–11344, 2020. 2

[19] Hao Li, Hong Zhang, Xiaojuan Qi, Ruigang Yang, and Gao Huang. Improved techniques for training adaptive deep networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1891–1900, 2019. 3

[20] Lei Li, Yankai Lin, Deli Chen, Shuhuai Ren, Peng Li, Jie Zhou, and Xu Sun. Cascadebert: Accelerating inference of pre-trained language models via calibrated complete models cascade. *arXiv preprint arXiv:2012.14682*, 2020. 3

[21] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 2

[22] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022. 2

[23] Xiaoxiao Li, Ziwei Liu, Ping Luo, Chen Change Loy, and Xiaoou Tang. Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3193–3202, 2017. 3

[24] Xiaonan Li, Yunfan Shao, Tianxiang Sun, Hang Yan, Xipeng Qiu, and Xuanjing Huang. Accelerating bert inference for sequence labeling via early-exit. *arXiv preprint arXiv:2105.13878*, 2021. 3

[25] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020. 1, 2

[26] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022. 3

[27] Kaiyuan Liao, Yi Zhang, Xuancheng Ren, Qi Su, Xu Sun, and Bin He. A global past-future early exit method for accelerating inference of pre-trained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2013–2023, 2021. 3

[28] Junyang Lin, Rui Men, An Yang, Chang Zhou, Ming Ding, Yichang Zhang, Peng Wang, Ang Wang, Le Jiang, Xianyan Jia, et al. M6: A chinese multimodal pretrainer. *arXiv preprint arXiv:2103.00823*, 2021. 1, 2

[29] Junyang Lin, An Yang, Yichang Zhang, Jie Liu, Jingren Zhou, and Hongxia Yang. Interbert: Vision-and-language interaction for multi-modal pretraining. *arXiv preprint arXiv:2003.13198*, 2020. 2

[30] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Haotang Deng, and Qi Ju. Fastbert: a self-distilling bert with adaptive inference time. *arXiv preprint arXiv:2004.02178*, 2020. 3

[31] Yijin Liu, Fandong Meng, Jie Zhou, Yufeng Chen, and Jinan Xu. Faster depth-adaptive transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13424–13432, 2021. 3

[32] Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. *Advances in Neural Information Processing Systems*, 34:28092–28103, 2021. 3

[33] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019. 2

[34] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10437–10446, 2020. 2

[35] Priyadarshini Panda, Abhronil Sengupta, and Kaushik Roy. Conditional deep learning for energy-efficient and enhanced pattern recognition. In *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 475–480. IEEE, 2016. 3

[36] Mary Phuong and Christoph H Lampert. Distillation-based training for multi-exit architectures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1355–1364, 2019. 3

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2

[38] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. 1

[39] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020. 3

[40] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021. 3

[41] Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. *arXiv preprint arXiv:2207.07061*, 2022. 5

[42] Tal Schuster, Adam Fisch, Tommi Jaakkola, and Regina Barzilay. Consistent accelerated inference via confident adaptive transformers. *arXiv preprint arXiv:2104.08803*, 2021. 2, 4

[43] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. 2

[44] Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2464–2469. IEEE, 2016. 3, 5

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[46] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022. 1, 2, 3, 4, 6

[47] Wenhui Wang, Hangbo Bao, Li Dong, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *arXiv preprint arXiv:2111.02358*, 2021. 2

[48] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021. 1, 2, 3

[49] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*, 2019. 2, 4, 5

[50] Ji Xin, Rodrigo Nogueira, Yaoliang Yu, and Jimmy Lin. Early exiting bert for efficient document ranking. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 83–88, 2020. 3

[51] Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. Deebert: Dynamic early exiting for accelerating bert inference. *arXiv preprint arXiv:2004.12993*, 2020. 1, 3, 4, 5, 6

[52] Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. Berxit: Early exiting for bert with better fine-tuning and extension to regression. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: Main Volume*, pages 91–104, 2021. 1, 3

[53] Haiyang Xu, Ming Yan, Chenliang Li, Bin Bi, Songfang Huang, Wenming Xiao, and Fei Huang. E2e-vlp: end-to-end vision-language pre-training enhanced by visual learning. *arXiv preprint arXiv:2106.01804*, 2021. 2

[54] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Bin Xiao, Ce Liu, Lu Yuan, and Jianfeng Gao. Unified contrastive learning in image-text-label space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19163–19173, 2022. 2

[55] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. Crossing the format boundary of text and boxes: Towards unified vision-language modeling. *arXiv preprint arXiv:2111.12085*, 2021. 2

[56] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 2

[57] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588, 2021. 1, 2

[58] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16793–16803, 2022. 2

[59] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13041–13049, 2020. 2

[60] Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. Bert loses patience: Fast and robust inference with early exit. *Advances in Neural Information Processing Systems*, 33:18330–18341, 2020. 3, 6