Neurocomputing 482 (2022) 60–72

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

ACORT: A compact object relation transformer for parameter efficient image captioning

Jia Huei Tan [a], Ying Hua Tan [a], Chee Seng Chan [a,*], Joon Huang Chuah [b]

[a] CISiP, Faculty of Computer Science and Information Technology, Universiti Malaya, 50603 Kuala Lumpur, Malaysia
[b] Faculty of Engineering, Universiti Malaya, 50603 Kuala Lumpur, Malaysia

ARTICLE INFO

Article history:
Received 20 May 2021
Revised 21 December 2021
Accepted 23 January 2022
Available online 29 January 2022
Communicated by Zidong Wang

Keywords:
Image captioning
Deep network compression
Deep learning

ABSTRACT

Recent research that applies Transformer-based architectures to image captioning has resulted in state-of-the-art image captioning performance, capitalising on the success of Transformers on natural language tasks. Unfortunately, though these models work well, one major flaw is their large model sizes. To this end, we present three parameter reduction methods for image captioning Transformers: Radix Encoding, cross-layer parameter sharing, and attention parameter sharing. By combining these methods, our proposed ACORT models have 3.7× to 21.6× fewer parameters than the baseline model without compromising test performance. Results on the MS-COCO dataset demonstrate that our ACORT models are competitive against baselines and SOTA approaches, with CIDEr score ⩾126. Finally, we present qualitative results and ablation studies to demonstrate the efficacy of the proposed changes further. Code and pre-trained models are publicly available at https://github.com/jiahuei/sparse-image-captioning.

1. Introduction

Recent years have seen stunning advances in the performance of Deep Neural Networks (DNNs) on sequence modelling tasks, particularly natural language processing, understanding, and generation. One of the main drivers behind these successes is the Transformer model [1], which forgoes recurrence in favour of a fully-attentional feed-forward architecture. Besides enabling parallelisation through time during training, Transformers with their residual connections can be stacked to form a multi-layer model. By increasing model depth and width, more powerful models can be trained, and task performance can be improved [2].

Riding on the successes of Transformers, recent research extending Transformer-based architectures to image captioning have resulted in state-of-the-art (SOTA) captioning performance [3,4]. Notably, the Object Relation Transformer (ORT) [3] model, which combines the Faster R-CNN detector network from [5] with a pair of Transformer-based encoder and decoder, improves the SOTA performance by 8 CIDEr[1] score. This is then further improved by [4] with the Meshed-Memory Transformer.

Unfortunately, whilst these models provide good performance, one main shortcoming is their large model sizes. For example on MS-COCO [6], the Soft-Attention model [7] has only 11.9 M parameters, whereas the ORT model has 55.4 M parameters (a 4.7× increase). Furthermore, as the size of datasets grows larger, the vocabulary size increases as well, leading to large input and output embedding matrices which further exacerbates parameter inefficiency. This can impede real-time applications deployment in resource-constrained devices such as mobile and embedded devices. Moreover, large models also consume more memory during training and inference, reducing overall efficiency due to the communication overhead incurred. Finally, recent works on natural language modelling, generation and understanding also suggests that Transformer models are over-parameterised [8,9], a trend that is shared among other DNN models [10].

To this end, we address the aforementioned problems in this paper by introducing ACORT – A Compact Object Relation Transformer architecture for parameter efficient image captioning. By incorporating three parameter reduction methods, ACORT models can achieve performances that are on par with the standard ORT model but with significantly fewer parameters. Firstly, Radix Encoding [11] is utilised to drastically reduce the size of the embedding matrices, allowing vocabulary size to grow without affecting model compactness. Secondly, cross-layer parameter sharing is used to decouple the strong correspondence between model depth and model size, allowing more layers to be stacked

* Corresponding author.
E-mail addresses: tanjiahuei@siswa.um.edu.my (J.H. Tan), tanyinghua@siswa.um.edu.my (Y.H. Tan), cs.chan@um.edu.my (C.S. Chan), jhchuah@um.edu.my (J.H. Chuah).

[1] Consensus-based Image Description Evaluation (CIDEr) is a widely-used metric for measuring caption quality via the level of consensus between ground-truth and generated captions.

https://doi.org/10.1016/j.neucom.2022.01.081
0925-2312/

without increasing parameter count and vice versa. Thirdly, attention parameter sharing is utilised to reduce the parameter count of the multi-head attention module, thereby further improving overall parameter efficiency (see Fig. 1).

As a result of these optimisations, an ACORT model with the same configuration as ORT has 3.7× fewer parameters yet has slightly better performance, while the smallest ACORT model has 21.6× fewer parameters with minimal performance degradation. In summary, the core contributions of this work are twofold. Firstly, we propose ACORT, A Compact ORT model with reduced vocabulary size and parameter count (up to 13× smaller). Secondly, we demonstrate the effectiveness of ACORT on the MS-COCO [6] dataset. Our experiments show that even the smallest ACORT configuration can reach a CIDEr score ⩾126, which is comparable to many SOTA approaches (see Section 5.3).

2. Related Works

2.1. Image Captioning

Image captioning is the task of generating descriptive captions given an image. It is a problem that requires scene understanding and natural language generation capabilities. While early works relied on hand-designed templates and rule-based systems, recent works have seen consistent performance improvements driven by the use of DNNs. Here we discuss several notable related works; see [12,13] for surveys.

The end-to-end differentiable encoder-decoder architecture that directly generates a caption given an image was popularised by [14,15], and inspired many later works [16,17]. Subsequently, attention was utilised to condition the caption generation process on salient visual features from the image encoder [18,19]. Attributes have also been used to inject semantic information into the caption generation process [20,21]. Following that, [5] used an object detector as a form of hard-attention to generate image features from bounding boxes. This framework was then extended by replacing the Long-Short Term Memory (LSTM) decoder with Transformers [1]. Such works include [22,23,4,24], and ORT [3], which is the baseline used in this work. More details on ORT are given in Section 3. At the same time, much work has gone into reinforcement learning, which has enabled the use of non-differentiable caption metrics as optimisation objectives [25,26].

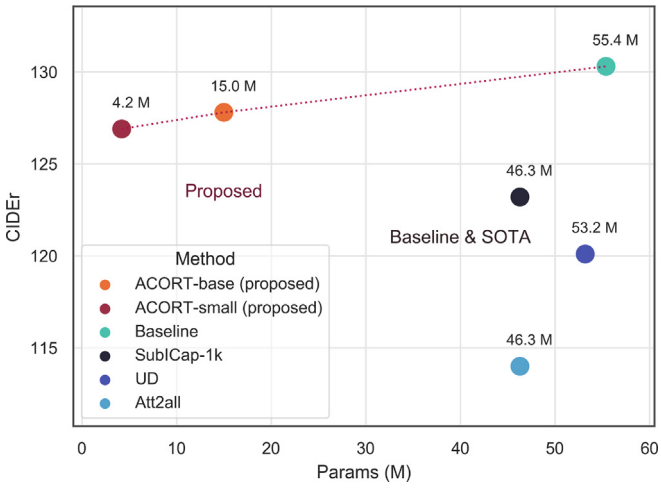

Fig. 1. The proposed ACORT models can achieve competitive CIDEr scores on MS-COCO against baseline and state-of-the-art models despite having 3.7× to 13.2× fewer parameters. See Table 5 for details.

While these methods have been effective in improving SOTA test performance, there has been comparatively little effort put into reducing model size, which is the primary motivation for this paper.

2.2. Parameter Efficient Transformers

Parameter sharing techniques have been utilised to great effect on various Transformer architectures.

[27] proposed the Universal Transformer, which performs recurrent stacking of layers, effectively performing cross-layer sharing. Such recurrent stacking is also used by RSNMT [8]. [28] suggested deep sequence models produce hidden activations that converge towards some fixed point, and thus proposed the DEQ model that finds these equilibrium points via root-finding to form an infinite depth, weight-tied feedforward network. Subsequently, [29] proposed the ALBERT model, a lightweight Transformer model with cross-layer parameter sharing for pretraining and finetuning on natural language understanding tasks. Motivated by AlBERT, [30] employed low-rank approximation of Transformer weights via singular value decomposition for video representation learning. More works utilising cross-layer sharing include [31,32]. Going a step further, [33] performed model-level sharing in which the encoder and the decoder of the Transformer translation model share the same set of weights. In a similar fashion, the Levenshtein Transformer [34] shares the Transformer backbone of three policy classifiers for natural language generation and refinement tasks.

Attention parameter sharing methods were also applied to Transformers. [35] shares the self-attention probability maps and the cross-attention outputs across adjacent Transformer layers. [36] proposed the Reformer, which utilised query-key weight sharing to enable the use of locality-sensitive hashing for attention computation.

Despite their remarkable effectiveness, weight sharing strategies have remained largely unexplored for image captioning Transformer models. To this end, we investigate the performance of two different sharing strategies on image captioning, namely cross-layer sharing and attention parameter sharing.

2.3. Neural Architecture Search, Network Pruning, Quantisation

Methods based on Neural Architecture Search (NAS) have been applied to automatically search for compact architectures that outperform human-designed architectures on performance and efficiency [37,38]. A notable example is the NASNet architecture for image classification, which achieves SOTA accuracy across different levels of computational cost [39]. However, these works have yet to include parameter sharing into their design space. Furthermore, such methods usually involve optimising a separate controller network using reinforcement learning which complicates the training procedure [40,41].

On the other hand, techniques such as network pruning, quantisation and distillation can also be used to effectively reduce the number of network parameters [42,43]. While these methods have been used to compress Transformers [44–46], they are orthogonal to this work. In principle, these techniques can be applied on top of our proposed compact model to achieve further parameter and disk space savings.

2.4. Parameter Efficient Image Captioning

[47] designed a compact image captioning network which combined SqueezeNet [48] and LightRNN [49] in a Soft-Attention [7] framework. By using LightRNN with factorised word embeddings, the size of the model is reduced substantially. Meanwhile, [11]

proposed the COMIC model, which utilised Radix Encoding, down-projections with weight sharing and multi-head attention to improve the parameter efficiency of Soft-Attention. In the work, Radix Encoding is used to factorise token embeddings across time steps, while weight-tied down-projections are used to reduce the model size further. [50] proposed the H-LSTM cell, which adds extra hidden layers to the vanilla LSTM cell. By doing so, H-LSTM can reduce the number of parameters by requiring fewer external stacked layers. However, their work derives the majority of parameter reduction from the use of the grow-and-prune method [51] rather than architectural modifications. Finally, [52] proposed the SubICap model, which uses SentencePiece tokenisation [53] instead of word-level tokenisation. Since SentencePiece performs subword segmentation, its vocabulary size can be made smaller, thus reducing model size.

In this paper, we focus on reducing the number of parameters in Transformers for image captioning. Similar to [52], the ORT architecture is chosen as the basis of our work. However, while [52] only targets the embedding matrices for parameter reduction, our methods perform weight reduction across the entire Transformer. As a result, our proposed models have significantly fewer parameters with comparable test performance. Performance comparisons are given in Section 5.2 and 5.3.

3. Object Relation Transformer: Revisited

Since our proposed ACORT model builds on the *Object Relation Transformer (ORT)* architecture proposed by Herdade et al. [3], we present a brief introduction of the ORT model before moving on to our ACORT model. Fig. 2 shows an overview of ORT.

The ORT model is an encoder-decoder architecture that extends the Up–Down model of [5]. The key innovation is the use of geometric attention rather than traditional soft-attention to incorporate spatial relationships between the detected objects into attention weights. The geometric attention module operates by multiplying the standard attention weights between two objects with a learned function of their relative location and scale. These geometric relations are encoded in the form of a spatial coordinate displacement vector along with width and height ratios between the objects.

Moreover, rather than just combining the Faster R-CNN detector network with an LSTM decoder, ORT employs an additional Transformer-based encoder network on top of the detector. The encoder is a 6-layer Transformer network. The R-CNN generated image feature vectors are fed into the encoder Transformer, but they are down-projected from 2,048 channels to 512 channels to fit the Transformer network's dimensionality. Embedded image features are then transformed by a sequence of multi-head scaled dot-product self-attention, multi-layer perceptron (MLP) with ReLU, and layer normalisation [56] layers with residual connections [57] inside the Transformer. These transformed features are then fed into the Transformer decoder's cross-attention module.

The decoder is similar to the encoder, but there are three main differences. First, the decoder's multi-head self-attention performs masked self-attention in order to achieve auto-regressive modelling. Second, the Transformer decoder conducts multi-head cross-attention between the encoder and the decoder in addition to multi-head self-attention. This cross-attention module receives the encoder output features and outputs a convex combination of the features as context vectors. Finally, positional embeddings are used to insert information about the relative positions of the token embeddings that the Transformer decoder receives as input.

4. ACORT: A Compact Object Relation Transformer

In this section, we present the overall design of ACORT and explain the reasoning behind the components used in ACORT. Quantitative and qualitative comparisons against the standard ORT baseline as well as SOTA approaches are provided in Section 5.

4.1. Radix Encoding

The original ORT formulation uses a standard architecture whereby a word embedding matrix is used to generate the embedding vector as model input, with a separate output projection matrix used to generate a probability distribution over all the output tokens. On a moderately-sized dataset such as MS-COCO [6],

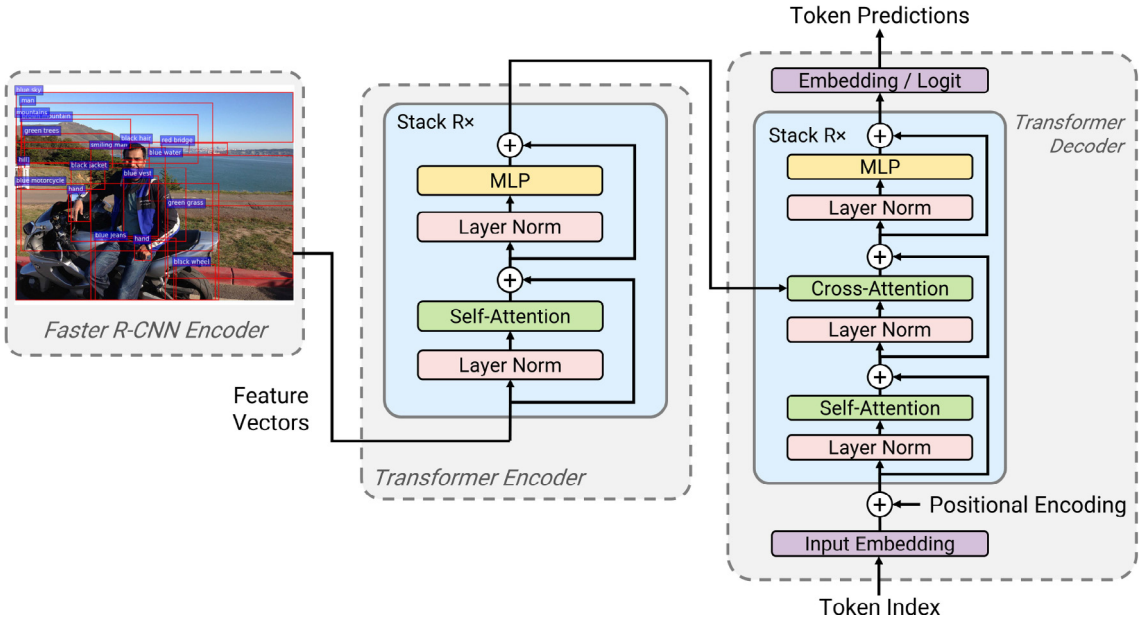

Fig. 2. Overview of the Object Relation Transformer (ORT) architecture [3]. Each blue box represents a single Transformer layer. Here, the output embedding layer is also known as the logit or pre-softmax projection layer [54]. Figure adapted from [5,55].

the use of word-level tokenisation scheme generates a large vocabulary size and inflated embedding matrices. Typically, these embeddings matrices have sizes that grow in lockstep with the size of the datasets used to train the models. This is due to the fact that that distribution of words in natural language usually follows a Zipfian distribution [58].

To this end, Radix Encoding was proposed in [11] as a way to sidestep the aforementioned issue. The method operates by representing each token index i as a combination of indices $\hat{i}$. Technically, the encoded indices $\hat{i}$ for regular tokens can be computed following Eq. (1)–(3) below. The word indices are assigned based on word frequency, such that the most common word is assigned to the first index and the least common word is assigned to the last index.

$$\hat{i}_j = \lfloor i / v^j \rfloor \bmod v \quad \text{for} j \in \{0, \dots b, d-1\} \text{and} i \notin \{i_{BOS}, i_{EOS}\}, \tag{1}$$

$$\hat{i} = v \quad \text{for} i = i_{BOS}, \tag{2}$$

$$\hat{i} = v + 1 \quad \text{for} i = i_{EOS}, \tag{3}$$

where v is the Radix base, and $\lfloor \cdot \rfloor$ denotes the *floor*$(\cdot)$ operator converting floating-point values to the nearest integer that is less than or equal to the input. The process is illustrated in Fig. 3. The pseudocode for Radix Encoding is given in Algorithm 1.

Following the equations, token-splitting is performed on all the tokens in the vocabulary, except for the two special tokens: the <BOS> symbol to start the caption generation process, and the <EOS> symbol to signal the end of the generation process. Doing so allows the original vocabulary V_o of size v^d to be compactly encoded into a new vocabulary V_e of size $v+2$, where d is the number of Radix tokens required to encode each original token. Note that d and v are inversely proportional to each other, where for a given vocabulary size $|V_o|$, a smaller Radix base v will lead to a larger d. In practice, $d = 2$ is able to provide good performance while maintaining model compactness.

By employing the encoding scheme on top of the word-level tokenisation, we can compress the original embedding matrices from a size of $v^d \times r$ to $(v+2) \times r$. This leads to an exponential reduction in embedding sizes at a factor of d, in which $|V_e| \ll |V_o|$. There is also no need to alter model code as one can simply run inference using greedy or beam search (or other inference methods) as usual and apply post-processing on the output tokens.

Algorithm 1: Radix Encoding

Input: Training corpus D, Radix base v
Output: Word-to-radix mapping V_e
count ← List of words and its corresponding frequency in D
count ← Sort *count* by word frequency in descending order
  V_o ← Assign an index to each word in *count* ⇒The original word vocabulary
V_e ← Empty dictionary ⇒The Radix vocabulary
for each(word, i) in V_o**do**
Compute $(\hat{i}_0, \dots b, \hat{i}_j)$ from i and v using Eq. (1)
$V_e[word] \leftarrow (\hat{i}_0, \dots b, \hat{i}_j)$
end
$V_e[BOS] \leftarrow v$ ⇒Refer to Eq. (2)
$V_e[EOS] \leftarrow v+1$ ⇒Refer to Eq. (3)
returnV_e

4.2. Cross-Layer Sharing

In addition to Radix Encoding, cross-layer sharing is utilised as another effective method for parameter reduction in ACORT. A Transformer with cross-layer sharing applied will have a subset or all of its layers share parameters. In the case where the number of layers is larger than the number of independent layers, this can lead to significant parameter reductions. For clarity, the layer configurations are denoted as follows: A 6-layer Transformer with 3 independent layers (i.e. 3 layers shared across 2 layers each) is denoted as `(0,0,1,1,2,2)`, a 6-layer Transformer with 2 independent layers is denoted as `(0,0,0,1,1,1)` or `(0x3,1x3)`, and a 2-layer Transformer with 2 independent layers is denoted as `(0,1)`. These configurations are illustrated in Fig. 4.

As outlined in Section 2.2, there are many ways to share Transformer parameters across its layers. A notable example is the ALBERT model, which shares all parameters across its layers [29]. This means that a 6-layer ALBERT will have a layer-sharing configuration of `(0x6)`. While this aggressive sharing scheme yields a model that is very lightweight, it is not the ideal configuration for image captioning (see Table 7).

In this paper, we explore the application of cross-layer sharing in a group-wise fashion, such that there is more than one independent layer in the Transformer. Specifically, we tested various layer-sharing configurations with the number of independent layers ranging from one to five. Empirical result in Section 5.5 shows that a model with two independent layers can provide good performance and parameter count reduction.

4.3. Attention Parameter Sharing

While cross-layer sharing is an effective inter-layer parameter sharing strategy, further weight savings can be derived by leveraging intra-layer parameter sharing. Specifically, each of the multi-head scaled dot-product attention modules in a Transformer layer contains three weight matrices W_Q, W_K, W_V that can be potentially shared. Fig. 5 shows two types of parameter sharing scheme explored in this paper: *Share-kV* and *Share-QK*.

In a nutshell, *Share-QK* operates by sharing the parameters of the weight matrices W_Q and W_K. Likewise, *Share-kV* performs parameter sharing on the weight matrices W_K and W_V. In addition to reducing learnable parameters, these weight sharing schemes can reduce computation cost by allowing some linear projection operations to be skipped. This is achieved by reusing the output from the shared intermediate projection layer, illustrated in Fig. 6.

In self-attention, all three inputs to the attention module – query, key, value – originates from the same source tensor. In other words, the inputs to self-attention are identical among each other ($Q \equiv K \equiv V$). Thus for self-attention, both *Share-QK* and *Share-kV* allow for intermediate projection reuse. However, this is not the case for cross-attention where query comes from the decoder while key and value come from the encoder. Hence for cross-attention, only *Share-kV* is able to skip the computation of value projection by directly reusing the projected key (key-value reuse).

4.4. Final Proposed Setup

Our proposed model ACORT incorporates all of the aforementioned parameter reduction strategies, namely Radix Encoding, cross-layer parameter sharing, and attention parameter sharing.

Original Vocabulary		Radix Encoding, $v = 25$			Radix Encoding, $v = 256$	
a →	0	a →	0	0	a →	0 0
on →	1	on →	0	1	on →	0 1
...	...	...	...	...	...	...
towing →	2024	towing →	3	5 24	towing →	7 232
star →	2025	star →	3	6 0	star →	7 233
...	...	...	...		...	...
<EOS> →	9960	<EOS> →	26		<EOS> →	257

Fig. 3. Example of original vocabulary mapping (left), encoded Radix base-25 vocabulary (middle) and encoded Radix base-256 vocabulary (right).

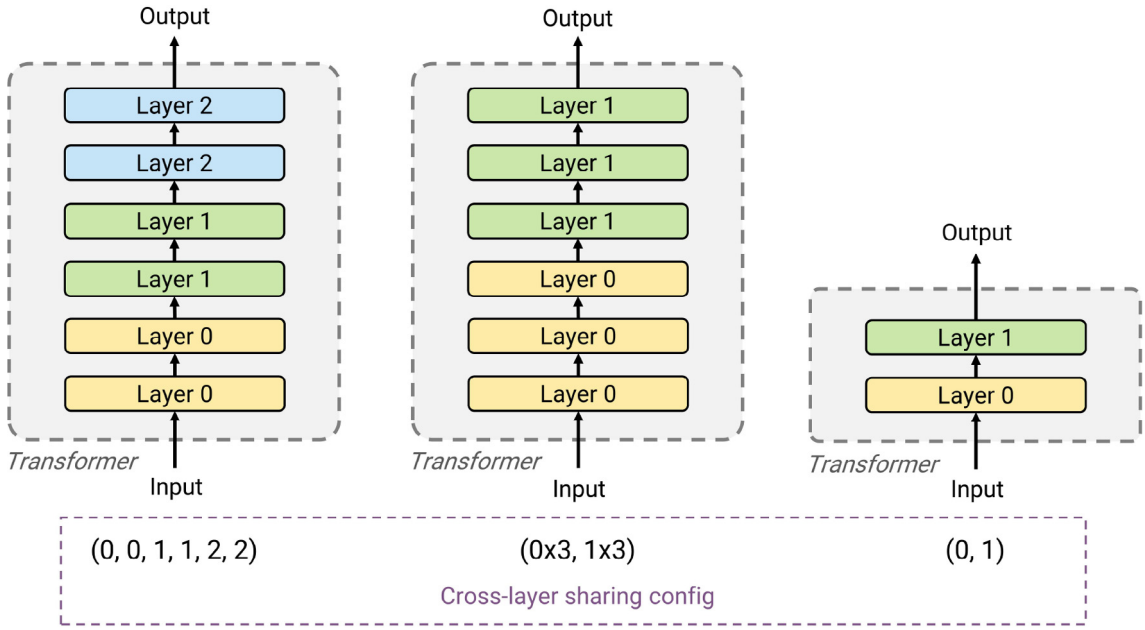

Fig. 4. Example configurations of cross-layer sharing in a Transformer. `(0,0,1,1,2,2)` means that there are 6 layers in total and sharing is applied every two layers. `(0x3,1x3)` is similar except sharing is applied every three layers. `(0,1)` means that there are 2 layers in total and no sharing is applied.

The differences in model size and setup compared to the standard ORT baseline are presented in Table 1.

Radix Encoding: ACORT utilises a base number of $v = 768$, which provided a good balance of performance and parameter efficiency (see Section 5.3). Other Radix base values are also explored in Table 6, from $v = 256$ up to $v = 1024$.

Cross-layer sharing: ACORT uses `(0x3,1x3)` layer-sharing for the *-base* and *-small* configurations. For the *-xsmall* configuration, `(0x2)` layer-sharing is used to yield maximum parameter reduction. We also evaluated the performance of an ALBERT style layer-sharing scheme `(0x6)`, denoted as the *ACORT-base-AL* configuration. Table 9.

Attention parameter sharing: ACORT uses *Share-kV* as the attention parameter sharing method, as it shows good performance compared to both *Share-QK* and the baseline (see Table 10). Moreover, it has the slight advantage of allowing key-value reuse for both self-attention and cross-attention.

As a result of these optimisations, the ACORT-base configuration has 3.7× fewer parameters than that of ORT-base (55.4 M → 15.0 M). The ACORT-small configuration has 13.2× fewer parameters than ORT-base (55.4 M → 4.2 M), and 4.0× fewer parameters than ORT-small (16.7 M → 4.2 M). The smallest ACORT configuration has 21.6× fewer parameters than ORT-base (55.44 M → 2.57 M), yet it is able to reach a CIDEr score of 126, which is comparable to many SOTA approaches (see Section 5.3). Meanwhile, even though the ORT *-small* and *-xsmall* configurations are compact, their performances are less ideal.

In terms of baseline models, both the *-small* and *-xsmall* configurations have the same number of layers as the *-base* model, but with narrower layers. In addition, we also included the *-base-2* and *-base-4* configurations with the same width as the *-base* model, but with fewer layers. Performance comparison between baselines and ACORT is given in Section 5.2.

5. Experiments

In this section, we present empirical results on the performance of our ACORT model, and provide comparisons against both base-

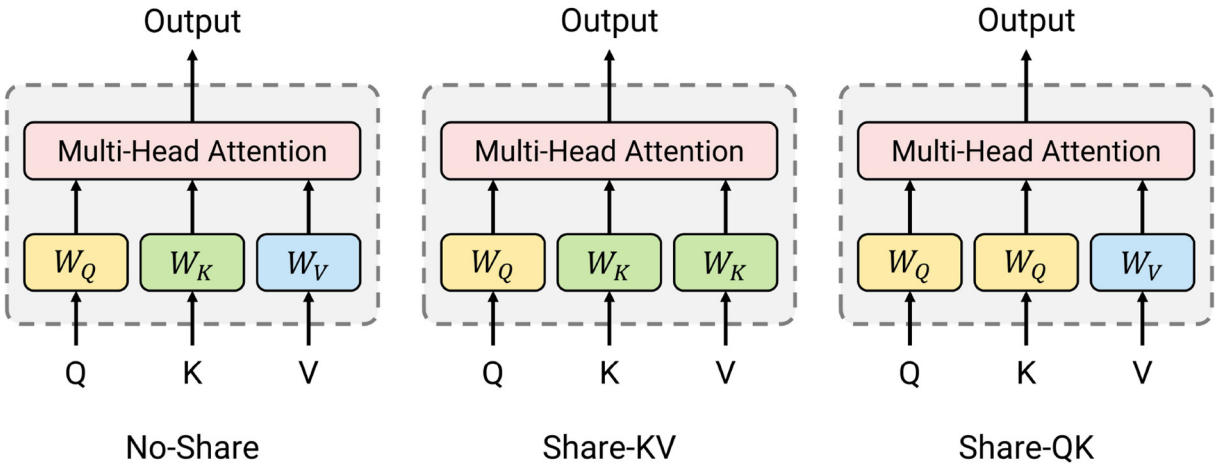

Fig. 5. Example of attention parameter sharing. *No-Share* is the standard baseline attention. *Share-kV* shares W_K and W_V, while *Share-QK* shares W_Q and W_K.

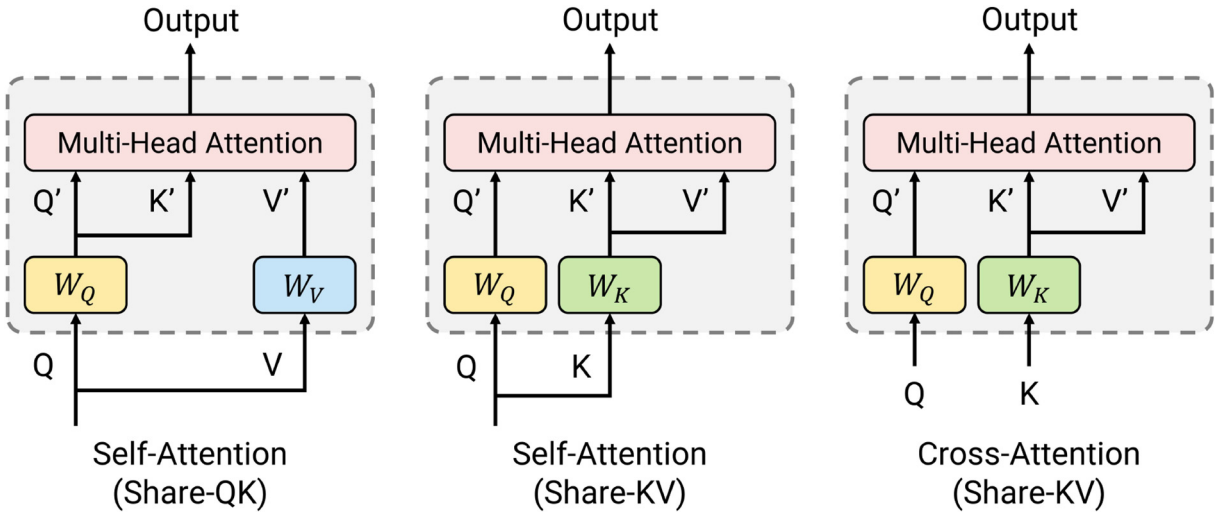

Fig. 6. When attention parameter sharing is used, some projection computations can be skipped by reusing the output of the shared projection layer.

Table 1
The configurations of the ORT and ACORT models, unless otherwise stated.

Approach		Params (M)	Radix base v	Layer config	Att. param sharing	Hidden size	MLP size
ORT (baseline)	base	55.4				512	2,048
	small	16.7	-	(0, 1, 2, 3, 4, 5)	No-Share	256	1,024
	xsmall	4.1				104	416
	base-4	40.7	-	(0, 1, 2, 3)	No-Share	512	2,048
	base-2	26.0		(0, 1)			
ACORT	base	15.0	768	(0x3, 1x3)	Share-kV	512	2,048
	base-AL	8.4		(0x6)			
	small	4.2	768	(0x3, 1x3)	Share-kV	256	1,024
	xsmall	2.6		(0x2)			

line and SOTA approaches. We also provide extensive ablation studies on the contribution of each component in ACORT.

5.1. Experiment Setup

Our experiment setup closely follows ORT [3] to allow for meaningful quantitative comparisons.

Implementation and hyperparameters: We reuse the public implementations published by the authors[2] for both ACORT and baseline models. Following [3], Adam is utilised as the optimiser, with the "Noam" learning rate scheduler for cross-entropy training and the step learning rate schedule for SCST training [25].

Our SCST training utilised the recently proposed method by [4,59] for performing action space sampling and computing the reward baseline. Random search is used to generate multiple sampled captions for each image, and the baseline for each image is set to be the average of the caption scores.

Inference is performed using beam search without length normalisation, using the model checkpoint with the highest validation CIDEr score. Following [3], test set performance is obtained using

[2] https://github.com/yahoo/object_relation_transformer

Table 2
Test set performance comparison against the baseline ORT models. ACORT models provide competitive test performance with significantly fewer parameters when compared to baseline models.

Approach		Params (M)	MS-COCO test scores					
			B-1	B-4	M	R	C	S
ORT (baseline)	base	55.4	76.1	35.2	27.9	56.5	114.7	21.0
	base-4	40.7	76.4	35.3	27.8	56.4	114.0	21.1
	base-2	26.0	76.7	35.9	27.8	56.6	114.8	21.2
	small	16.7	76.1	35.1	27.5	56.2	113.0	20.6
	xsmall	4.1	74.9	32.7	26.2	54.9	104.4	19.1
ACORT	base	15.0	76.6	35.9	28.1	56.9	116.0	21.4
	base-AL	8.4	75.6	33.8	27.2	55.7	110.0	20.8
	small	4.2	76.0	34.6	27.4	56.0	112.5	20.7
	xsmall	2.6	75.1	33.7	27.1	55.6	108.7	20.3

Table 3
Uniqueness and average length of the generated captions on MS-COCO. Captions generated by ACORT contain 2.0% to 4.6% more unseen sentences that are ∼0.2 words longer.

Approach		Unique (%) [a]	Average word count
ORT (baseline)	base	61.2	9.52
	base-4	60.1	9.44
	base-2	60.6	9.38
	small	60.4	9.39
	xsmall	57.6	9.23
ACORT	base	63.2	9.59
	base-AL	71.7	9.73
	small	62.2	9.44
	xsmall	62.8	9.53

[a]Percentage of generated captions not found in training set.

beam size of 2 after cross-entropy optimisation; and beam size of 5 after SCST optimisation.

Datasets and metrics: Experiments are performed on MS-COCO [6], a popular English captioning dataset for benchmarking. The "Karpathy" split [14] is utilised, which assigns 5,000 images for validation, 5,000 for testing and the rest for training. Preprocessing of captions is done following [11]. Evaluation scores are obtained using the publicly available MS-COCO evaluation toolkit[3], which computes BLEU, METEOR, ROUGE-L, CIDEr and SPICE (B, M, R, C, S).

5.2. Baseline Comparison

5.2.1. Test performance

Table 2 shows the comparison between model size and the metric scores of our model against the baseline after cross-entropy (teacher-forcing) training. The configurations presented here are detailed in Section 4.4.

Overall, the performance of ACORT is on par with the various baseline configurations. Comparing the *-base* configurations, our ACORT-base model is able to slightly outperform ORT-base, ORT-base-4 and ORT-base-2 models, despite having 42% to 73% fewer parameters. It achieves relative improvements of 2.0% on BLEU-4, 1.1% on CIDEr, and 1.9% on SPICE compared to ORT-base. Interestingly, ORT-base-2 outperforms both ORT-base and ORT-base-4, suggesting that the original configuration might be overparameterised and could benefit from the regularisation effects of simply having fewer parameters.

In addition, ACORT-small can provide competitive test performance even with a large 92.4% reduction in parameter count (13×). Compared to ORT-base, its scores dropped by merely 1.7% on BLEU-4, 1.9% on CIDEr, and 1.4% on SPICE. When compared to ORT-xsmall with a similar parameter count, ACORT-small clearly outperforms it, with relative improvements of 5.4% on BLEU-4, 4.3% on CIDEr, and 1.9% on SPICE. At the same time, ACORT-xsmall with only 2.6 M parameters (38.0% reduction) also outperforms ORT-xsmall. While a considerable score gap exists between ACORT-xsmall and ORT-base, the CIDEr gap can be reduced significantly through the SCST training procedure (see Section 5.3). On the other hand, ACORT-base-AL with only one independent layer underperforms compared to both ACORT-base and ACORT-small, with score gaps of −2.3% on BLEU-4, −2.2% on CIDEr, and +0.5% on SPICE compared to ACORT-small.

One notable observation from our experiments is the training instability of the ORT-xsmall configuration. The model tends to diverge during training, and we had to repeat its training run twice to get a model that did not diverge early in the process. Other than that, all the other configurations had stable training runs, including the ACORT models.

5.2.2. Caption uniqueness and length

To further assess the quality of captions generated by the models, we calculated the captions' uniqueness and average word count. Results are given in Table 3. Once again, the ACORT models were able to produce good performance. In terms of both caption uniqueness and length, ACORT is able to outperform the baselines. ACORT-generated captions contain 1.0% to 14.1% more unseen sentences, and the average lengths are around 0.2 words longer.

5.2.3. Training and inference cost

We examine the training and inference costs of our ACORT models as well as the baseline models. We measured the GPU memory required for the training process using figures reported by *torch.cuda.memory_reserved()* at the end of the first 1,000 training steps. Similarly, the training speedups were computed based on the time taken to perform the first 1,000 model updates. Inference speedups were calculated based on the time taken to generate captions on the entire test set with a batch size of 50 and a beam size of 2, following the setup described in Section 5.1. All results were obtained using a Titan Xp GPU.

Table 4 shows that the ACORT models consume less GPU memory during training due to their lower parameter counts. This is especially true for ACORT-xsmall, which is both extremely compact with 2.6 M parameters and lightweight with only 2 layers in total. These are also reflected in the training speedups enjoyed by the ACORT models, with speedups ranging from 1.2× to 4.3×. In terms of inference speeds, the ACORT models suffered slightly at 0.8× speedup compared to the baselines. This can be attributed to the increased sequence length incurred by the use of Radix Encoding. Further discussion is provided in Section 6. However, we note that our ACORT-xsmall model can still provide a 1.2×

[3] https://github.com/salaniz/pycocoevalcap

Table 4
Training and inference cost comparison against the baseline ORT models. ACORT models consume less GPU memory during training with 20% to 432% faster training speeds compared to ORT-base.

Approach		Params (M)	GPU Memory (↓)		Relative Speedup (↑)	
			GB	Relative	Training	Inference
ORT (baseline)	base	55.4	2.54	1.0×	1.0×	1.0×
	base-4	40.7	2.00	0.8×	1.4×	1.1×
	base-2	26.0	1.22	0.5×	2.5×	1.3×
	small	16.7	1.53	0.6×	1.1×	0.9×
	xsmall	4.1	0.87	0.3×	1.1×	0.9×
ACORT	base	15.0	2.36	0.9×	1.2×	0.8×
	base-AL	8.4	2.29	0.9×	1.3×	1.1×
	small	4.2	1.50	0.6×	1.4×	0.8×
	xsmall	2.6	0.57	0.2×	4.3×	1.2×

Table 5
Test set performance of the proposed model ACORT along with SOTA methods on MS-COCO. ACORT models provide good performance-to-size ratio, with parameter reductions of up to 18× when compared to SublCap-1 k.

Approach		Params (M) [a]	MS-COCO test scores (single-model)							
			B-1	B-2	B-3	B-4	M	R	C	S
Att2all [25]	46.3 [b]	-	-	-	34.2	26.7	55.7	114.0	-	
UD [5]		53.2 [b]	79.8	-	-	36.3	27.7	56.9	120.1	21.4
ORT [3]		55.4 [b]	80.5	-	-	38.6	28.7	58.4	128.3	22.6
AoANet [18]		-	80.2	-	-	38.9	29.2	58.8	129.8	22.4
M2 [4]		-	80.8	-	-	39.1	29.2	58.6	131.2	22.6
X-Transformer [19]		-	80.9	65.8	51.5	39.7	29.5	59.1	132.8	23.4
LightRNN [47] [c]		11.8	67.5	46.5	32.1	22.6	22.0	-	-	-
H-LSTM [50] [c]		-	71.9	-	-	-	-	-	95.4	-
ComIC-256 [11]		4.0	75.3	-	-	34.4	-	-	105.0	19.0
SublCap-1 k [52]		46.3	79.5	-	-	37.1	29.8	58.2	123.2	23.0
ACORT	base	15.0	79.3	64.1	49.9	38.2	28.4	57.8	127.8	21.6
	small	4.2	79.7	64.4	50.2	38.5	28.1	57.6	126.9	21.3
	xsmall	2.6	79.0	63.5	49.3	37.7	28.0	57.3	126.0	21.2

[a]Excluding image encoder. [b]Calculated based on reimplementation. [c]Teacher-forcing training only.

Table 6
The effect of different Radix Encoding bases. Overall, Radix Encoding with $v = 768$ provides good performance while maintaining embedding compactness.

Radix base v	Params (M)		MS-COCO validation scores					
	Embeddings	Total	B-1	B-4	M	R	C	S
Word (baseline)	10.3	55.4	75.5	33.9	27.6	56.2	111.0	20.6
1,024	1.1	46.2	75.1	33.8	27.8	56.3	111.9	20.8
768	0.8	46.0	75.2	33.3	27.7	55.9	111.7	20.8
512	0.5	45.7	74.9	33.6	27.9	56.2	111.9	20.7
256	0.3	45.5	75.2	33.3	27.6	56.1	110.6	20.6

Table 7
The effects of varying numbers of independently parameterised layers applied to both the encoder and the decoder. Configuration `(0x3,1x3)` achieves the closest metric scores when compared to the baseline model while remaining reasonably compact.

# independent layers	Layer share config	Params (M)	MS-COCO validation scores					
			B-1	B-4	M	R	C	S
6 (baseline)	(0, 1, 2, 3, 4, 5)	55.4	75.5	33.9	27.6	56.2	111.0	20.6
4	(0, 0, 0, 1, 2, 3)	40.7	75.9	33.5	27.4	55.8	110.6	20.6
3 (successive)	(0, 0, 1, 1, 2, 2)	33.4	75.9	33.9	27.5	56.2	111.0	20.5
3 (symmetric)	(0, 1, 2, 2, 1, 0)	33.4	75.8	33.3	27.0	55.6	110.5	20.3
2	(0x3, 1x3)	26.0	76.1	33.7	27.1	55.8	110.9	20.4
1	(0x6)	18.7	76.2	33.4	26.9	55.6	109.4	20.2

inference speedup compared to ORT-base, with minimal performance degradation after SCST optimisation is completed (see Table 5).

All in all, these results show that the proposed modifications are effective at reducing model parameter count while maintaining test performance.

5.3. State-of-the-Art Comparison

Table 5 demonstrates the test set performance of ACORT models compared to state-of-the-art approaches. The compact image captioning models are separated into the second group in the table.

Table 8
The performance of networks with only 2 independent layers but with varying degree of layer reuse. The 6-layer model with `(0x3,1x3)` configuration provides the best performance relative to the baseline.

# layers	Layer share config	Params (M)	MS-COCO validation scores					
			B-1	B-4	M	R	C	S
6 (baseline)	(0, 1, 2, 3, 4, 5)	55.4	75.5	33.9	27.6	56.2	111.0	20.6
2 (2 ind.)	(0, 1)	26.0	75.6	33.7	27.3	55.9	110.5	20.5
6 (2 ind.)	(0x3, 1x3)	26.0	76.1	33.7	27.1	55.8	110.9	20.4
12 (2 ind.)	(0x6, 1x6)	26.0	75.9	33.4	26.9	55.7	108.9	20.2

Table 9
The effects of applying cross-layer sharing to encoder only, decoder only, or both. Sharing both the encoder and the decoder provides the most parameter reduction while maintaining performance.

# independent layers	Layer share config	Params (M)	MS-COCO validation scores					
			B-1	B-4	M	R	C	S
Baseline	(0, 1, 2, 3, 4, 5)	55.4	75.5	33.9	27.6	56.2	111.0	20.6
Share encoder only	(0x6)	39.7	75.4	33.6	27.4	56.1	110.0	20.5
Share decoder only	(0x6)	34.4	75.8	33.5	26.9	55.6	109.0	20.4
Share both	(0x6)	18.7	76.2	33.4	26.9	55.6	109.4	20.2

Table 10
The effect of different attention parameter sharing configurations. Since there seems to be no significant performance difference among the various configurations, *Share-kV* is chosen to allow for key-value reuse.

Attention params sharing	Params (M)		MS-COCO validation scores					
	Attention	Total	B-1	B-4	M	R	C	S
No-Share (baseline)	18.9	55.4	75.5	33.9	27.6	56.2	111.0	20.6
Share-kV (encoder)	17.3	53.9	75.6	33.9	27.5	56.1	111.0	20.4
Share-kV (decoder)	15.8	52.3	75.6	33.9	27.4	56.1	111.1	20.7
Share-kV	14.2	50.7	75.6	33.8	27.6	56.2	111.2	20.7
Share-QK	14.2	50.7	75.7	34.0	27.7	56.3	111.9	20.7

Compared to compact image captioning models, ACORT models provide better performance while having smaller model sizes. ACORT is able to outperform all of the existing approaches on most metrics, with the lone exception of *SubICap-1 k*. By comparing ACORT against COMIC-256, the performance improvements derived from better image features (Faster R-CNN) and sequence decoder (Transformer) are apparent: ACORT-xsmall achieves +9.6% on BLEU-4, +20.0% on CIDEr, and +11.6% on SPICE. At the same time, ACORT-xsmall is 36% smaller than COMIC-256 (1.47 M decrease in parameters).

Against the ORT-based *SubICap-1 k*, ACORT-base has 67.6% fewer parameters (3×), ACORT-small has 90.9% fewer parameters (11×), while ACORT-xsmall has 94.5% fewer parameters (18×). Even with such large reductions in model sizes, ACORT is still able to provide good performance. Specifically, ACORT-base achieves +3.0% on BLEU-4 and +3.7% on CIDEr, ACORT-small achieves +3.8% on BLEU-4 and +3.0% on CIDEr, whereas ACORT-xsmall achieves +1.6% on BLEU-4 and +2.3% on CIDEr.

ACORT models are also competitive compared to standard SOTA models. The smallest ACORT model mostly outperforms the UD (Up–Down) model, with better scores on BLEU-4, METEOR, ROUGE and CIDEr. Even compared with the best models – AoANet, M2, and X-Transformer – all three ACORT models are able to produce competitive metric scores considering their small parameter counts.

In short, ACORT models provide good performance-to-size ratios, demonstrating the effectiveness of the proposed modifications outlined in Section 4.

5.4. Qualitative Results

In this section, we present some examples of the generated captions from ACORT along with the baseline ORT-base model. The captions are generated on the validation set using a beam size of 5.

Fig. 7 showcases some images where the captions generated by the ACORT models are accurate and descriptive relative to the ORT-base model. Despite their small sizes, both ACORT models still managed to convey specific and relevant details regarding the image content. For image (1), ACORT correctly described the action of the player as "hitting the ball" rather than holding it. For image (2), all the captions are accurate but ORT-base suffered from "bad endings" due to SCST optimisation [60]. For image (3), the fence was mentioned by ACORT but ignored by ORT-base. For image (4), the number of motorcycles is correctly specified as one by ACORT instead of two.

Fig. 8 showcases images where the captions generated by the ACORT models contain mistakes relative to the ORT-base model. This is to be expected, as the ACORT and baseline models have similar metric scores. From the captions, the presence of bad endings can again be seen. For image (1), even though the bus is correctly described as being "blue and yellow" by ACORT, it is incorrectly described as "driving down a street"; whereas the baseline caption is more accurate. For image (2), ACORT is unable to provide detailed descriptions of the elephants, compared to ORT-base which mentioned both the adult and the baby. Similarly for image (3) and (4), the baseline captions are more detailed, with mentions of "a pile of books" and "a yellow bus".

Overall, the captions generated by ACORT are still on par with those generated by the baseline ORT model. This shows that caption quality is maintained despite the reductions in parameter count.

5.5. Ablation Studies

In this section, we provide an extensive analysis of the design choices and architectural modifications made to the baseline ORT model [3]. By isolating each of the modifications and comparing them to the baseline, their contribution towards the performance and compactness of the final model ACORT can be better quantified. Unless stated otherwise, any parameter sharing configuration or modification is applied to both the encoder and decoder. All the

	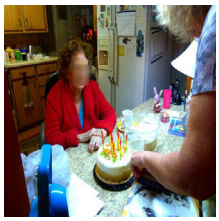		
ORT-base			
a man standing on a tennis court holding a ball	a woman sitting at a table with a birthday cake with candles on a	a brown bear sitting in the grass	two motorcycles parked on the side of a road
ACORT-base			
a man hitting a tennis ball on a tennis court	a woman sitting at a table with a birthday cake with candles	a brown bear sitting in the grass next to a fence	a motorcycle parked on the side of a road
ACORT-small			
a man hitting a tennis ball on a tennis court	a woman sitting at a table with a cake with candles	a brown bear sitting in the grass next to a fence	a motorcycle parked on the side of a road
ACORT-xsmall			
a man hitting a tennis ball on a tennis court	a woman sitting in front of a birthday cake with candles	a brown bear sitting in the grass in a field	a motorcycle parked on the side of a road

Fig. 7. Examples of images where the captions generated by the ACORT models are accurate and descriptive relative to the baseline.

			
ORT-base			
a bus parked in a parking lot at night	an adult elephant and a baby elephant standing in a field	a cat sitting on top of a pile of books	a yellow double decker bus driving down a street
ACORT-base			
a blue and yellow bus driving down a street	two elephants standing next to a baby elephant in a field	a cat sitting on top of a book	a double decker bus driving down a street
ACORT-small			
a bus driving down a street at night	two elephants standing in a field in a	a cat sitting on top of a book on a	a double decker bus driving down a street
ACORT-xsmall			
a bus parked in a parking lot at night	two elephants and a baby elephant standing in a field	a cat sitting on top of a pile of books	a double decker bus driving down a street with

Fig. 8. Examples of images where the ACORT models made mistakes.

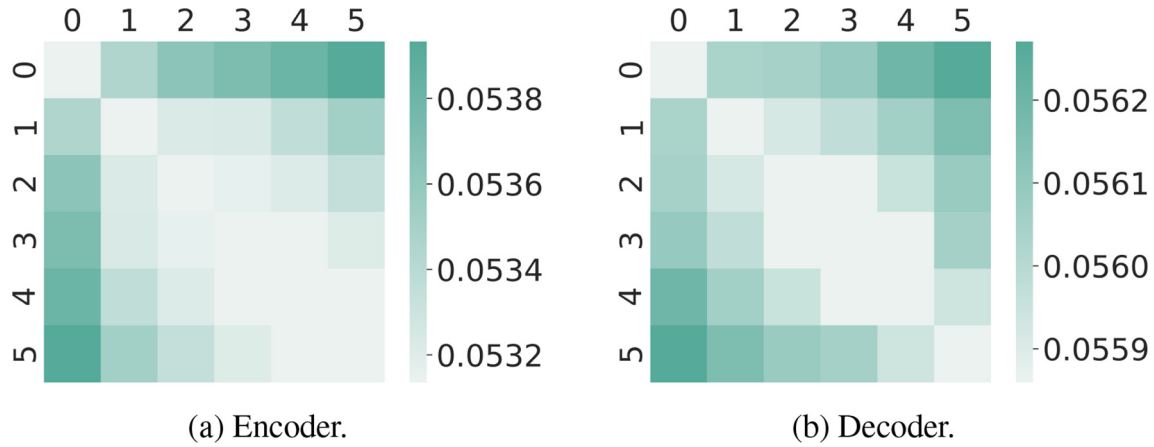

Fig. 9. The root mean squared distance for all layer pairs in a standard 6-layer ORT model. Minimum distance in each plot is clipped at its second lowest value. In general, the first layer is more distinct compared to the rest of the layers; layers 2 to 6 are closer to neighbouring layers.

metric scores presented in this section are obtained using greedy decoding on the MS-COCO validation set.

5.5.1. Radix encoding

Table 6 demonstrates the validation performance and parameter reductions that can be achieved using Radix Encoding. Across the board from base of $v = 256$ to $v = 1024$, the performance of the models is comparable to that of the baseline. This result shows that the Transformer model can effectively learn the strict dependencies between tokens imposed by Radix Encoding, despite its lack of recurrent connection.

Furthermore, the sizes of the embedding matrices are reduced significantly up to 97.4% (10.3 M → 0.3 M). At the same time, the performance degradation at the extreme end is merely 0.36% for CIDEr (111.0 → 110.6). All in all, this result suggests that there exists substantial redundancy in the embeddings matrices, which can be removed without affecting overall performance [61,11]. The work of [58] also found that performing embedding factorisation along with weight sharing can lead to performance improvements in language modelling.

5.5.2. Cross-layer sharing

Fix the number of layers, vary the number of independent layers: Table 7 summarises the performance of baseline networks with varying numbers of uniquely or independently parameterised layers. Cross-layer sharing is able to drastically reduce the number of parameters from 55.4 M to just 18.7 M, i.e. a 66.2% reduction. Furthermore, the performance of the `(0x6)` model with just one independent layer shared across 6 layers is still comparable to that of baseline, with CIDEr score of 109.4 versus 111.0 and BLEU-1 score of 76.2 versus 75.5. In general, while the broad trend remains that models with more parameters perform slightly better, the parameter-performance trade-off is still favourable to layer-shared models.

While Table 7 provides the performance data of various layer sharing configurations, we seek to explain the reasons behind the performance differences. To this end, we measured how the parameter distribution of a layer differs from each other in the baseline 6-layer ORT model. The results are given in Fig. 9, obtained by performing L_2 normalisation on the last axis[4] of the layer parameters, flattening and concatenating them, and computing the pair-wise mean squared distance.

The figure shows that the first layer – which processes the input embeddings – is more distinct compared to the rest of the layers. This largely supports the observation that sharing configurations where the first layer is shared more `(0,0,0,1,2,3)` or shared with the last layer `(0,1,2,2,1,0)` suffered from worse performance. Configuration `(0x6)` also suffered from being the worst performer, although that can be attributed in part to its lower parameter count. In addition, the similarity data also show that layers 2 to 6 are closer to neighbouring layers. This suggests that configurations where successive layers are shared will be more performant. Indeed, configurations `(0,0,1,1,2,2)` and `(0x3,1x3)` achieve the closest metric scores when compared to the baseline model.

Fix the number of independent layers, vary the number of layers: Table 8 demonstrates the performance where the number of independent layers is fixed but each layer is reused across varying numbers of layers in a recurrent fashion similar to [27]. Here, all the layer-shared models have two independent layers. From the results, even though all three models have very different computation costs – from 2 layers `(0,1)` to 12 layers `(0x6,1x6)` – they all end up with similar performance. This indicates that repeated computation using the same layer is not an efficient use of compute, with diminishing returns past 3 layer repetitions. Finally, it can be noted that the 6-layer model with `(0x3,1x3)` configuration provides the best performance relative to the baseline.

Apply layer sharing to encoder only, decoder only, or both: Table 8 shows that applying layer sharing to the encoder only while leaving the decoder intact produced less performance drop compared to sharing the decoder across all metrics except BLEU-1. Interestingly, applying layer sharing to both the encoder and decoder did not seem to worsen overall performance when compared to the decoder-only sharing scheme, with almost no discernible difference in terms of metric scores. However, the number of parameters vastly favoured sharing both the encoder and decoder, with the resulting network having 66% fewer parameters than the baseline. Overall, sharing both the encoder and the decoder provides the best performance to parameter count trade-off.

5.5.3. Attention parameter sharing

Table 10 shows the impact of various parameter sharing configurations. All the attention-shared models are performing surprisingly well despite the reduction in attention parameters. This is especially true for the *Share-QK* model, as it is able to outperform the baseline model across all metrics. We hypothesise that this is due to the increased regularisation from weight sharing. While the *Share-kV* model performs slightly inferior to *Share-QK*, it has a minor advantage in terms of compute cost (as mentioned in Section 4.3). Finally, there seems to be no significant performance

[4] Normalisation across first or all axes produce similar results.

difference whether the sharing is applied to encoder only, decoder only, or to both. It can be noted that sharing the decoder's attention parameters provides larger parameter reduction than the encoder, as the decoder contains cross-attention layers absent from the encoder.

6. Limitations and Future Work

One of the limitations of this work is the number of hyperparameters introduced. In particular, cross-layer parameter sharing introduced a wide range of possible layer-sharing configurations. While traversing all the possible hyperparameter combinations can be expensive, we believe the comprehensive baseline comparison and ablation study provided in Section 5.2 and 5.5 can mitigate this shortcoming partially. Alternatively, efficient hyperparameter optimisation (HPO) methods [62,41] can be used to further search the hyperparameter space. However, since this work focuses on demonstrating the effectiveness of the proposed parameter reduction techniques rather than pursuing raw performance, we leave the use of HPO techniques to future work.

Besides that, the Radix Encoding method utilised performs token factorisation across time steps, leading to longer sequence lengths. In the future, we wish to incorporate additional neural encoding layers to compress and consolidate tokens across time steps. Alternatively, an ALBERT-style token factorisation could be used. At the output layer, strategies such as [63] and CNN-softmax [64] can be used to achieve model size reduction.

On the other hand, while the various parameter reduction techniques presented in this work are only applied to a Transformer-based model (ORT), all of them are in principle compatible with other types of image captioning models. For example, cross-layer parameter sharing can potentially be applied to a multi-layer Recurrent Neural Network (RNN) decoder [5] in order to share weights across the RNN layers. Such decoder can also be outfitted with the *Share-kV* cross-attention module as described in Section 4.3. Finally, Radix Encoding can and has been successfully used together with an RNN-based model [11]. Similarly, models with convolutional decoder [17] can employ cross-layer parameter sharing to share weights across convolutional layers, while simultaneously utilising *Share-kV* cross-attention and Radix Encoding. The exploration of these architectures is left as future work.

7. Conclusion

We have presented ACORT – A Compact Object Relation Transformer architecture for parameter efficient image captioning. ACORT models can achieve performances that are on par with the standard ORT model (CIDEr score $\geqslant$126) but with 3.7$\times$ to 21.6$\times$ fewer parameters. This is achieved via the incorporation of three parameter reduction methods: Radix Encoding, cross-layer parameter sharing, and attention parameter sharing. Our results on the MS-COCO dataset demonstrates that the proposed ACORT-base and ACORT-small models are capable of achieving metric scores that are competitive against baselines and SOTA approaches. Finally, qualitative results and ablation studies are also presented to further demonstrate the effectiveness of the proposed modifications. Together with the public release of model code and pre-trained checkpoints, we hope that these findings can spur research interest in the image captioning community.

CRediT authorship contribution statement

Jia Huei Tan: Conceptualization, Methodology, Software, Data curation, Investigation, Visualization, Validation, Writing - original draft, Writing - review & editing. **Ying Hua Tan:** Validation, Writing - review & editing. **Chee Seng Chan:** Writing - review & editing, Supervision, Project administration. **Joon Huang Chuah:** Writing - review & editing, Supervision, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: NIPS, 2017, pp. 5998–6008.
[2] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, Workshop on EMC2, NeurIPS.
[3] S. Herdade, A. Kappeler, K. Boakye, J. Soares, Image captioning: Transforming objects into words, in: NeurIPS, 2019, pp. 11137–11147.
[4] M. Cornia, M. Stefanini, L. Baraldi, R. Cucchiara, Meshed-Memory Transformer for Image Captioning, in: IEEE/CVF CVPR, 2020, pp. 10578–10587.
[5] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, L. Zhang, Bottom-up and top-down attention for image captioning and visual question answering, in: IEEE CVPR, 2018, pp. 6077–6086.
[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: Common objects in context, in: ECCV, 2014, pp. 740–755.
[7] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: ICML, 2015, pp. 2048–2057.
[8] R. Dabre, A. Fujita, Recurrent stacking of layers for compact neural machine translation models, AAAI 33 (01) (2019) 6292–6299.
[9] S. Mehta, M. Ghazvininejad, S. Iyer, L. Zettlemoyer, H. Hajishirzi, DeLighT: Very Deep and Light-weight Transformer, ICLR.
[10] G. Hinton, O. Vinyals, J. Dean, Distilling the Knowledge in a Neural Network, NIPS Deep Learning and Representation Learning Workshop.
[11] J.H. Tan, C.S. Chan, J.H. Chuah, COMIC: Toward A Compact Image Captioning Model With Attention, IEEE TMM 21 (10) (2019) 2686–2696.
[12] M.Z. Hossain, F. Sohel, M.F. Shiratuddin, H. Laga, A comprehensive survey of deep learning for image captioning, ACM CSUR 51 (6) (2019) 1–36.
[13] X. Liu, Q. Xu, N. Wang, A survey on deep neural network-based image captioning, The Visual Computer 35 (3) (2019) 445–470.
[14] A. Karpathy, L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: IEEE CVPR, 2015, pp. 3128–3137.
[15] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: A neural image caption generator, in: IEEE CVPR, 2015, pp. 3156–3164.
[16] Y.H. Tan, C.S. Chan, Phrase-based image caption generator with hierarchical LSTM network, Neurocomputing 333 (2019) 86–100.
[17] R. Li, H. Liang, Y. Shi, F. Feng, X. Wang, Dual-CNN: A Convolutional language decoder for paragraph image captioning, Neurocomputing 396 (2020) 92–101.
[18] L. Huang, W. Wang, J. Chen, X.-Y. Wei, Attention on attention for image captioning, in: IEEE/CVF ICCV, 2019, pp. 4634–4643.
[19] Y. Pan, T. Yao, Y. Li, T. Mei, X-linear attention networks for image captioning, in: IEEE/CVF CVPR, 2020, pp. 10971–10980.
[20] H. Fang, S. Gupta, F. Iandola, R.K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J.C. Platt, et al., From captions to visual concepts and back, in: IEEE CVPR, 2015, pp. 1473–1482.
[21] Q. Wu, C. Shen, P. Wang, A. Dick, A. van den Hengel, Image captioning and visual question answering based on attributes and external knowledge, IEEE TPAMI 40 (6) (2017) 1367–1381.
[22] L. Zhou, Y. Zhou, J.J. Corso, R. Socher, C. Xiong, End-to-end dense video captioning with masked transformer, in: IEEE CVPR, 2018, pp. 8739–8748.
[23] J. Yu, J. Li, Z. Yu, Q. Huang, Multimodal transformer with multi-view visual representation for image captioning, IEEE TCSVT 30 (12) (2019) 4467–4480.
[24] Z. Yu, N. Han, Accelerated masked transformer for dense video captioning, Neurocomputing 445 (2021) 72–80.
[25] S.J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, V. Goel, Self-Critical Sequence Training for Image Captioning, in: IEEE CVPR, 2017, pp. 1179–1195.
[26] H. Chen, G. Ding, S. Zhao, J. Han, Temporal-difference learning with sampling baseline for image captioning, in: AAAI, 2018, pp. 6706–6713.
[27] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, Ł. Kaiser, Universal transformers, ICLR.
[28] S. Bai, J.Z. Kolter, V. Koltun, Deep Equilibrium Models, in: NeurIPS, Vol. 32, 2019, pp. 690–701.
[29] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, ICLR.
[30] S. Lee, Y. Yu, G. Kim, T. Breuel, J. Kautz, Y. Song, Parameter Efficient Multimodal Transformers for Video Representation Learning, arXiv preprint arXiv:2012.04124.
[31] D. Sachan, G. Neubig, Parameter Sharing Methods for Multilingual Self-Attentional Translation Models, in: Proceedings of the Conference on Machine Translation: Research Papers, 2018, pp. 261–271.
[32] M. Reid, E. Marrese-Taylor, Y. Matsuo, Subformer: Exploring Weight Sharing for Parameter Efficiency in Generative Transformers, arXiv preprint arXiv:2101.00234.

[33] Y. Xia, T. He, X. Tan, F. Tian, D. He, T. Qin, Tied transformers: Neural machine translation with shared encoder and decoder, AAAI 33 (01) (2019) 5466–5473.
[34] J. Gu, C. Wang, J. Zhao, Levenshtein transformer, in: NeurIPS, Vol. 32, 2019, pp. 11179–11189.
[35] T. Xiao, Y. Li, J. Zhu, Z. Yu, T. Liu, Sharing attention weights for fast transformer, in: IJCAI, 2019, pp. 5292–5298.
[36] N. Kitaev, Ł. Kaiser, A. Levskaya, Reformer: The efficient transformer, ICLR.
[37] X. Dong, Y. Yang, Network pruning via transformable architecture search, in: NeurIPS, 2019, pp. 760–771.
[38] X. Dong, Y. Yang, Searching for a robust neural architecture in four gpu hours, in: IEEE/CVF CVPR, 2019, pp. 1761–1770.
[39] B. Zoph, V. Vasudevan, J. Shlens, Q.V. Le, Learning transferable architectures for scalable image recognition, in: IEEE CVPR, 2018, pp. 8697–8710.
[40] H. Pham, M. Guan, B. Zoph, Q. Le, J. Dean, Efficient neural architecture search via parameters sharing, in: ICML, 2018, pp. 4095–4104..
[41] X. Dong, M. Tan, A.W. Yu, D. Peng, B. Gabrys, Q.V. Le, AutoHAS: Efficient hyperparameter and architecture search, ICLR, Workshop Track Proceedings.
[42] M. Lin, R. Ji, Y. Zhang, B. Zhang, Y. Wu, Y. Tian, Channel Pruning via Automatic Structure Search, in: IJCAI, 2020, pp. 673–679.
[43] W. Wen, Y. He, S. Rajbhandari, M. Zhang, W. Wang, F. Liu, B. Hu, Y. Chen, H. Li, Learning Intrinsic Sparse Structures within Long Short-Term Memory, ICLR.
[44] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, I. Titov, Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned, in: Proceedings of the Annual Meeting of the ACL, 2019, pp. 5797–5808.
[45] G. Prato, E. Charlaix, M. Rezagholizadeh, Fully Quantized Transformer for Machine Translation, in: EMNLP: Findings, 2020, pp. 1–14.
[46] P. Ganesh, Y. Chen, X. Lou, M.A. Khan, Y. Yang, D. Chen, M. Winslett, H. Sajjad, P. Nakov, Compressing Large-Scale Transformer-Based Models: A Case Study on BERT, arXiv preprint arXiv:2002.11985.
[47] S.N. Parameswaran, Exploring Memory and Time Efficient Neural Networks for Image Captioning, in: NCVPRIPG, Springer, 2017, pp. 338–347..
[48] F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <) 0.5 MB model size, arXiv preprint arXiv:1602.07360.
[49] X. Li, T. Qin, J. Yang, T.-Y. Liu, LightRNN: Memory and computation-efficient recurrent neural networks, in: NIPS, Vol. 29, 2016, pp. 4385–4393..
[50] X. Dai, H. Yin, N. Jha, Grow and Prune Compact, Fast, and Accurate LSTMs, IEEE Transactions on Computers 69 (3) (2020) 441–452.
[51] X. Dai, H. Yin, N. Jha, NeST: A neural network synthesis tool based on a grow-and-prune paradigm, IEEE Transactions on Computers 68 (10) (2019) 1487–1497.
[52] N. Sharif, M. Bennamoun, W. Liu, S.A.A. Shah, SubICap: Towards Subword-informed Image Captioning, in: IEEE/CVF WACV, 2021, pp. 3540–3549..
[53] T. Kudo, J. Richardson, SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing, in: EMNLP: System Demonstrations, 2018, pp. 66–71.
[54] O. Press, L. Wolf, Using the Output Embedding to Improve Language Models, in: Proceedings of the Conference of the EACL: Volume 2, Short Papers, 2017, pp. 157–163.
[55] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, ICLR..
[56] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450.
[57] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE CVPR, 2016, pp. 770–778.
[58] A. Baevski, M. Auli, Adaptive input representations for neural language modeling, ICLR.
[59] R. Luo, A Better Variant of Self-Critical Sequence Training, arXiv preprint arXiv:2003.09971.
[60] T. Guo, S. Chang, M. Yu, K. Bai, Improving Reinforcement Learning Based Image Captioning with Natural Language Prior, in: EMNLP, 2018, pp. 751–756..
[61] K. Shi, K. Yu, Structured Word Embedding for Low Memory Neural Network Language Model, in: Interspeech, 2018, pp. 1254–1258.
[62] J. Lorraine, P. Vicol, D. Duvenaud, Optimizing millions of hyperparameters by implicit differentiation, in: AISTATS, 2020, pp. 1540–1552.
[63] W. Chen, D. Grangier, M. Auli, Strategies for Training Large Vocabulary Neural Language Models, in: Proceedings of the Annual Meeting of the ACL (Volume 1: Long Papers), 2016, pp. 1975–1985.
[64] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, Y. Wu, Exploring the limits of language modeling, arXiv preprint arXiv:1602.02410.

Jia Huei Tan is currently working as an AI/Machine Learning Engineer in 3D Perception at AkiraKan Limited, while also pursuing a Ph.D. degree at the Faculty of Computer Science and Information Technology, Universiti Malaya (UM), Malaysia. He received a B.Eng. (Hons.) degree in Telecommunication Engineering from UM in 2016. His research interests include deep learning for computer vision and natural language understanding.

Ying Hua Tan received her B.E. degree in Mechatronics Engineering from UCSI University, Malaysia in 2013. She is currently working towards Ph.D. in University of Malaya, Malaysia. Her research interests include deep learning in computer vision and natural language processing.

Chee Seng Chan is currently an Associate Professor with the Faculty of Computer Science and Information Technology, Universiti Malaya, Kuala Lumpur, Malaysia. From 2020 till 2022, he is seconded to the Ministry of Science and Technology and Innovation (Malaysia) as an undersecretary. His research interests include computer vision and machine learning with focus on scene understanding. He is also interested in the interplay between language and vision: generating sentential descriptions about complex scenes. Dr. Chan was the Founding Chair for the IEEE Computational Intelligence Society Malaysia Chapter, the Organising Chair for the Asian Conference on Pattern Recognition (2015), the General Chair for the IEEE Workshop on Multimedia Signal Processing (2019), and IEEE Visual Communications and Image Processing (2013). He is a Chartered Engineer registered under the Engineering Council, U.K. He was the recipient of several notable awards, such as Young Scientist Network-Academy of Sciences Malaysia in 2015 and the Hitachi Research Fellowship in 2013.

Joon Huang Chuah received the B.Eng. (Hons.) degree from the Universiti Teknologi Malaysia, the M.Eng. degree from the National University of Singapore, and the M.Phil. and Ph.D. degrees from the University of Cambridge. He is currently an Associate Professor and Head of VIP Research Group at the Department of Electrical Engineering, Faculty of Engineering, University of Malaya. He is a Chartered Engineer registered under the Engineering Council, U.K., and also a Professional Engineer registered under the Board of Engineers, Malaysia. He was the Honorary Treasurer of IEEE Computational Intelligence Society (CIS) Malaysia Chapter and the Honorary Secretary of IEEE Council on RFID Malaysia Chapter. He is the Vice Chairman of the Institution of Engineering and Technology (IET) Malaysia Network. In addition, he is a Fellow and the Honorary Secretary of the Institution of Engineers, Malaysia (IEM) and a Committee Member of the Board of Engineers Malaysia (BEM). He serves in the Governing Board of ASEAN Federation of Engineering Organisations as Honorary Treasurer. His main research interests include image processing, computational intelligence, IC design, and scanning electron microscopy.