

VisualGPT: Data-efficient Image Captioning by Balancing Visual Input and Linguistic Knowledge from Pretraining

Jun Chen¹ Han Guo² Kai Yi¹ Boyang Li³ Mohamed Elhoseiny¹

¹King Abdullah University of Science and Technology

²Carnegie Mellon University ³Nanyang Technological University

{jun.chen,kai.yi,mohamed.elhoseiny}@kaust.edu.sa
hanguo@cs.cmu.edu, boyang.li@ntu.edu.sg

Abstract

The ability to quickly learn from a small quantity of training data widens the range of applications of machine learning. In this paper, we propose a data-efficient image captioning model, VisualGPT, which leverages the linguistic knowledge from a large pretrained language model (LM). A crucial challenge is to balance between the use of visual information in the image and prior linguistic knowledge acquired from pretraining. We designed a novel self-resurrecting encoder-decoder attention mechanism to quickly adapt the pretrained LM as the language decoder on a small amount of in-domain training data. The proposed self-resurrecting activation unit produces sparse activations but is not susceptible to zero gradients. When trained on 0.1%, 0.5% and 1% of MSCOCO [35] and Conceptual Captions [56], the proposed model, VisualGPT, surpasses strong image captioning baselines. VisualGPT outperforms the best baseline model by up to 10.8% CIDEr on MS COCO and up to 5.4% CIDEr on Conceptual Captions. To the best of our knowledge, this is the first work that improves data efficiency of image captioning by utilizing LM pretrained on unimodal data. Our code is available at: <https://github.com/Vision-CAIR/VisualGPT>.

1. Introduction

Image captioning [28, 63, 24, 12, 22] is a prominent example of cross-modal reasoning, requiring accurate understanding of the visual content and precise expression of that understanding in natural language. The task has been established for novel applications such as helping people with impaired vision to understand their surroundings [7, 11] and generating medical imaging for human physicians [33, 10].

However, most of the recent performance gains in image captioning relies on large-scale image-caption corpora such as MS COCO [35] or Conceptual Captions [56]. For

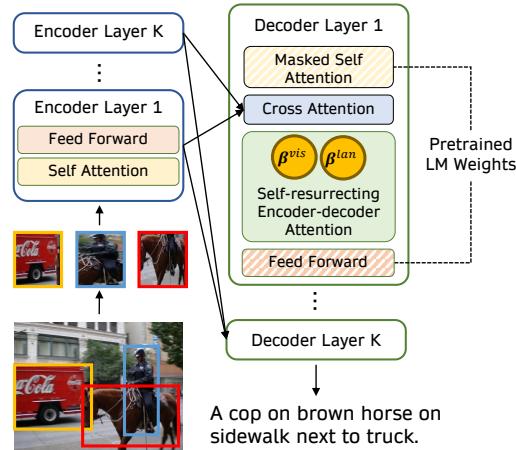


Figure 1. Our VisualGPT model transfers the knowledge from a pretrained language model to the caption decoder. A self-resurrecting encoder-decoder attention is designed to connect the multi-level visual features and caption decoder.

instance, MS COCO contains approximately one million human-written captions. Manually creating captions for such large datasets requires significant time and effort. On the other hand, semi-automatic approaches for collecting image-caption pairs from the Internet, as used by Conceptual Captions [56], may generate incorrect or undesirable training data even after multiple rounds of data cleaning; data crawled from the Internet are unlikely to cover highly specific domains such as computerized tomography (CT) scans. Thus, the availability of training data limits the range of objects and scenes that image captioning systems can reliably describe [1]. Improved data efficiency of image captioning will allow practitioners to quickly curate sufficient amount of data and establish systems that can describe rare objects in specific domains.

In this paper, we investigate the data efficiency problem for image captioning. This problem is distinct from the novel object captioning problem [20, 1], which relies

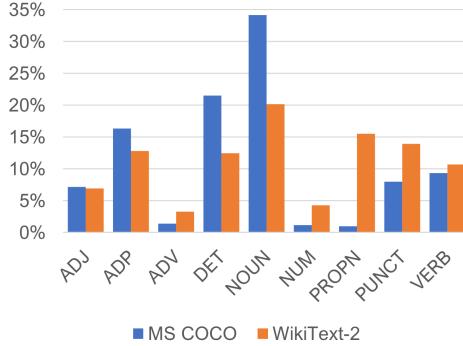


Figure 2. Comparison of the part-of-speech distributions of the MS COCO and WikiText-2 datasets [43]. We use the spacy parser and show only the most important categories.

on abundant in-domain data but zero out-of-domain data. Instead, we aim to improve the performance of image captioning systems trained on a small subset of *in-domain* data.

We propose to improve data efficiency by leveraging pre-trained language models (LMs) [15, 37, 30, 51], such as BERT [13], XLNet [65], and GPT [49, 50, 8]. Via self-supervised learning, these models acquire rich linguistic and semantic knowledge, which has been shown to inform downstream tasks in NLP [9, 17].

A challenge in utilizing pre-trained LMs is to bridge the gap between multi-modal data and the single-modal textual data the LMs are pretrained on. In Figure 2, we compare the part-of-speech distributions of MS COCO and WikiText-2 [43]. MS COCO employs 75% more nouns but 14% fewer verbs. This suggests that the MS COCO captions are biased toward descriptions of static objects rather than actions. As a result, effective use of pre-trained LMs in image captioning requires careful balancing of the linguistic knowledge acquired from pretraining and the visual input information.

Figure 1 shows the overall architecture of the proposed network, called VisualGPT. In the commonly used encoder-decoder architecture for image captioning, we initialize the parameters of the decoder from pre-trained LMs such as GPT-2 [50], whereas the encoder layers are randomly initialized. In addition, we propose a self-resurrecting attention mechanism that precisely balances the input from the visual encoder and the prior linguistic knowledge from the lower decoder layer. The proposed self-resurrecting attention mechanism can learn to ignore small magnitude inputs and produce sparse activations. Notably, the mechanism does not suffer from the zero gradient problem and can “turn on” an activation again after it has been zeroed out.

We evaluate our VisualGPT against several strong baseline models on 0.1%, 0.5% and 1% of the MS COCO dataset, and the experimental results demonstrate that our VisualGPT can easily outperform the baselines. We also conduct several ablative experiments to confirm the usefulness of pre-trained LMs and the proposed self-resurrecting

attention mechanism.

With this paper, we make the following contributions:

- We propose to investigate the data efficiency problem for image captioning and to borrow weights from pre-trained language models to initialize the decoder. Using only a small amount of in-domain training data, the proposed encoder-decoder quickly adapts network weights obtained from the textual modality to the cross-modal task of image captioning. To our knowledge, this is the first work that focuses on efficiently adapting large pre-trained language models for image captioning.
- We propose a novel encoder-decoder attention with a self-resurrecting activation unit (SRAU), which can learn to balance features from the visual and textual modalities. SRAU produces sparse activations while not being “trapped” in a zero-gradient region.
- We apply the proposed VisualGPT model to several small subsets of MS COCO and Conceptual Captions. In both automatic evaluation and human evaluation, VisualGPT surpasses several state-of-the-art baselines.

2. Related Work

Image Captioning. Numerous image captioning models have been proposed in the past few years. Earlier approaches generated the image caption template, and fill in the blanks with the outputs of object or attribute predictors [57, 66]. In contrast, modern approaches exploit the neural encoder-decoder architecture where an encoder network encodes the visual features and a decoder network generates the language description [62, 14, 23]. Visual features are may be represented by a grid of CNN features [63, 41] or image regions containing objects [3]. Graph neural networks have also been adopted to represent scene graphs or spatial relationships between objects [67, 64]. Recurrent networks [3, 4] and Transformer networks [31, 22, 12] are popular choices for the language decoder. Reinforcement learning enables model optimization with non-differentiable evaluation metrics [53, 36].

The novel image captioning problem treats the image captioning as a zero-shot learning where the captioning system is required to describe objects that do not exist in the training data. Lu *et al.* [42] propose a hybrid template-based method that fills the slots based on the object categories recognized by object detectors. Feng *et al.* [16] propose an unsupervised approach that trains via unpaired image-caption data and a visual concept detector. From a learning efficiency perspective, Kim *et al* [25] improve the data efficiency by borrowing the knowledge from unpaired image-caption data.

Self-supervised NLP Models. Self-supervised training of large neural networks on textual data proves to be an important technique in the creation of high-performance NLP models. Several self-supervision signals have been proposed, such as autoregressive language modeling and masked language modeling.

Autoregressive Language modeling is arguably one of the most fundamental task in NLP. The task is to predict the next word by conditioning on all preceding words. More formally, given a sequence of words w_1, \dots, w_N and a probability distribution p_θ parameterized by θ , the training objective is

$$\max_{\theta} \sum_{t=1}^N \log p_\theta(w_t | w_1, \dots, w_{t-1}) \quad (1)$$

Pretrained models using this objective include [6, 44] and the GPT series [50, 8, 49].

Another popular objective is masked language modeling, which predicts a randomly masked word in a textual sequence based on all other words. Given a random variable $Z \in \{1, \dots, N\}$, the training objective is

$$\max_{\theta} \mathbb{E}_Z [\log p_\theta(w_Z | w_1, \dots, w_{Z-1}, w_{Z+1}, \dots, w_N)] \quad (2)$$

Models using this objective include ELMo [47] and BERT-related methods [13, 29, 38].

In this paper, we propose a quick adaptation technique for network weights obtained using the language modeling objective. However, the technique is not specific to this type of self-supervision signal and can be applied to other models, as the masked LM objective can be easily converted to the LM objective by masking only the last word in the textual sequence.

Unlike neural networks pretrained on multimodal data (e.g., [48, 59, 58, 40, 32]), our method only requires a small amount of multimodal training data and focuses on adapting linguistic knowledge learned from the textual modality.

3. The VisualGPT Architecture

The VisualGPT model contains an image encoder and a caption decoder comprising K and M Transformer [60] layers, respectively. Given an image, we first extract objects in the image using an off-the-shelf object detection network. After that, we extract features from the detected bounding boxes and feed them into the image encoder. We denote the number of objects extracted as O and the dimension of hidden states in the Transformer layers as D . As such, the image encoder outputs a tensor I of dimension $D \times O \times K$. Conditioned on I , the caption decoder outputs words in the caption in an autoregressive manner. For the maximum caption length T , the decoder outputs a tensor C of dimension $D \times M \times T$. The output of the last decoder

layer are classified into sub-word tokens under Byte Pair Encoding (BPE) [55]. At layer m of the decoder, we use the self-resurrecting encoder-decoder attention mechanism to find the right balance between visual information I and the linguistic output $C[m - 1]$ from the immediate lower decoder layer. In the next few subsections, we will describe these network components in details.

3.1. Visual Encoder

The visual encoder consists of K Transformer layers, each of which contains a self-attention operation, a feed-forward operation, and addition-normalization operation. These components are described below.

The self-attention operation can be understood as encoding one elements in the input as a convex combination of other elements in the input. Let I_{k-1} denote the output of the encoder layer $i - 1$ and the input to the encoder layer k . We first linearly project the input to the query matrix \mathcal{Q} , key matrix \mathcal{K} , and value matrix \mathcal{V} .

$$\mathcal{Q} = W^q I_{k-1}, \mathcal{K} = W^k I_{k-1}, \text{ and } \mathcal{V} = W^v I_{k-1}, \quad (3)$$

where the matrices W^q , W^k , and W^v are learnable parameters for the j^{th} head at layer i . The output of the self-attention mechanism is a convex combination of the columns of \mathcal{V} .

$$f_{att}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) = \text{softmax} \left(\frac{\mathcal{Q}\mathcal{K}^\top}{\sqrt{D}} \right) \mathcal{V} \quad (4)$$

In standard multi-head attention, we utilize multiple sets of W^q , W^k , and W^v . The outputs of all heads are concatenated and linearly projected back to D dimensions. The output of the multi-head attention, I_k^{att} , is fed into a feed-forward neural network $\text{FFN}(\cdot)$, which is applied to each object feature separately and identically. It composes of two affine transformations and the GELU activation [21].

Finally, an encoder layer also contains residual connections and layer normalization, which are denoted by AddNorm . For an arbitrary input z and function $g(\cdot)$, the definition for AddNorm is

$$\text{AddNorm}(z, g(\cdot)) = \text{LayerNorm}(z + g(z)) \quad (5)$$

For simplicity, we just write $\text{AddNorm}(g(z))$ when there is no ambiguity. The final output of the encoder layer i is

$$I_i = \text{AddNorm}(\text{FFN}(\text{AddNorm}(I_k^{att}))) \quad (6)$$

The same encoding layer is repeated K times to form the complete encoder. The final output of the encoder contains the outputs of all layers, I_1, \dots, I_K , which form the tensor I of $D \times O \times K$ dimensions.

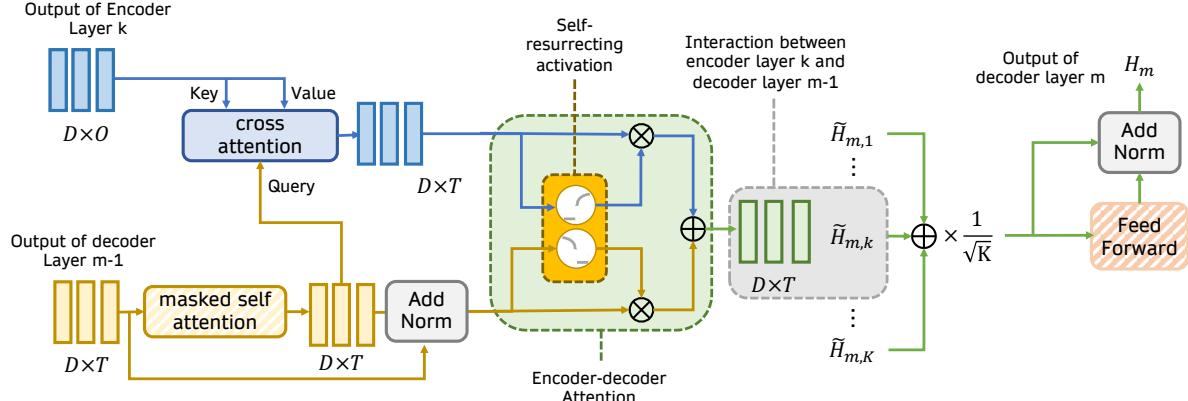


Figure 3. The architecture of one caption decoder layer. The diagram focuses on the interaction between the output of encoder layer k and the output of decoder layer $m - 1$, which yields the matrix $\tilde{H}^{m,k}$. After that, the final output of the layer, H^m , is derived from the sum of $\tilde{H}_{m,1}, \dots, \tilde{H}_{m,K}$. Striped boxes indicate network components initialized from pretrained weights.

3.2. Caption Decoder

We create the caption decoder by adapting network weights learned from the uni-modal language modeling task. A pretrained language model (LM), parameterized by θ , generates the next token w_t conditioned on the predecessor words w_1, \dots, w_{t-1} . To train the LM is to fit the conditional distribution $P_\theta(w_t|w_1, \dots, w_{t-1})$. In comparison, the caption decoder fits the conditional distribution $P_\theta(w_t|w_1, \dots, w_{t-1}, I)$, where I is the output of the image encoder. It may look simple as we add only one more term to the condition. However, in practice adding the term I completely changes the distribution because it requires the conciliation of information from two different modalities.

We hypothesize that the generation of visual words, such as “person”, “truck”, or “dog”, requires the model to rely on visual information. In contrast, the generation of determiners or connectives requires only linguistic knowledge. Ideally, we would like to exploit the massive amount of linguistic knowledge stored in the pretrained LM weights, while referring to the visual input only when required.

To achieve this goal, we design the decoder architecture, which contains a masked self-attention operation, a cross-attention operation, and an encoder-decoder attention with a self-resurrecting activation unit. Without loss of generality, we now describe these components in the decoder layer m . These components are also illustrated in Figure 3.

Masked Self-Attention. At decoder layer m , we apply masked self-attention, a standard component in Transformer-based language decoders, to the output of the decoder layer $m - 1$, which is denoted as H_{m-1} . In the decoding at time step t , we use a binary mask to prevent the self-attention operation from seeing any information at the time step $t + 1$ and beyond. The output of the masked self-attention is denoted as \dot{H}_m .

Cross-modality Attention. In the cross-modality attention, we linearly project the $\dot{H}_m \in \mathbb{R}^{D \times T}$ to the query matrix and the output of encoder layer k , $I_k \in \mathbb{R}^{D \times T}$ to both the key and value matrices. More formally, we apply the same f_{att} function defined in Equation 4,

$$\dot{I}_k = f_{att}(W^{dq} \dot{H}_m, W^{dk} I_k, W^{dv} I_k). \quad (7)$$

where W^{dq} , W^{dk} , and W^{dv} are trainable parameters.

Encoder-Decoder Attention. To quickly adapt a pretrained language model to a cross-modality task, it is crucial for the neural network to correctly employ the visual input and the linguistic knowledge acquired from pretraining at the right time. The visual information should take priority when generating common visual words, whereas the linguistic knowledge can contribute to connectives or uncommon words.

In order to balance the visual input \dot{I}_k and the linguistic input \dot{H}_m , we propose a new encoder-decoder attention. The balance is controlled by two gating matrices $B_m^{\text{vis}} \in [0, 1]^{D \times T}$ and $B_m^{\text{lan}} \in [0, 1]^{D \times T}$; they control the relative strengths of the visual input and linguistic input to decoder layer m . As such, we compute the interaction between the m^{th} decoder layer and the k^{th} encoder layer as

$$\tilde{H}_{m,k} = B_m^{\text{vis}} \otimes \dot{I}_k + B_m^{\text{lan}} \otimes \dot{H}_m \quad (8)$$

where \otimes denotes component-wise multiplication. The final output of the decoder layer, H_m , is computed as the sum of all encoder-decoder interaction.

$$\tilde{H}_m = \frac{1}{\sqrt{K}} \sum_{k=1}^K \tilde{H}_{m,k} \quad (9)$$

We introduce two techniques for computing B^{vis} and B^{lan} . The first soft gating technique computes them in pairs

using sigmoid activation.

$$B_m^{\text{vis}}[i, j] = \sigma(A[i, j]), \quad B_m^{\text{lan}}[i, j] = 1 - \sigma(A[i, j]) \quad (10)$$

where $B[i, j]$ denotes the i, j entry of matrix B . Here A is computed as an affine transformation of the two input matrices,

$$A = W_m^{\text{gate}}[\dot{I}_k; \dot{H}_m] + C_m^{\text{gate}}, \quad (11)$$

where $W_m^{\text{gate}} \in \mathbb{R}^{D \times 2D}$ and $C_m^{\text{gate}} \in \mathbb{R}^{D \times T}$ are trainable parameters, and $[I; H]$ denotes the concatenation of matrices I and H .

The final output of the decoder layer m is denoted as H_m and is computed using FFN and AddNorm.

$$H_m = \text{AddNorm}(\text{FFN}(\text{LayerNorm}(\tilde{H}_m))) \quad (12)$$

Self-Resurrecting Activation Unit. For the second method to compute B_m^{vis} and B_m^{lan} , we propose a novel paired activation function, which we call self-resurrecting activation unit (SRAU), defined as follows

$$\begin{aligned} \text{SRAU}(\alpha, \tau) &= [\sigma(\alpha) \mathbb{1}(\sigma(\alpha) > \tau); \\ &\quad 1 - \sigma(\alpha) \mathbb{1}(1 - \sigma(\alpha) > \tau)] \end{aligned} \quad (13)$$

The entries in the matrices B_m^{vis} and B_m^{lan} are computed from SRAU in pairs.

$$[B_m^{\text{vis}}[i, j]; B_m^{\text{lan}}[i, j]] = \text{SRAU}(A[i, j], \tau) \quad (14)$$

where τ is a predefined threshold hyperparameter and $\mathbb{1}(\cdot)$ is the indicator function, which returns 1 if the inner statement is true and 0 otherwise.

Figure 4 (left) plots the SRAU function. The SRAU contains two gates, $\sigma(\alpha)$ and $\sigma(1 - \alpha)$, which are complementary to each other. When one of the gate falls below the threshold τ , it is rectified to zero. This behavior suppresses small values and creates sparse activations, which may mitigate overfitting. However, when a gate variable is set to zero, it receives zero gradient and cannot be optimized via gradient descent. This is known as a “dead” gate and may impede proper optimization. It is worth noting that, in the design of SRAU, when $\tau < 0.5$, the two complementary gates cannot simultaneously receive zero gradient. In other words, if α receives zero gradient, $1 - \alpha$ can continue to be optimized. As such, the SRAU avoids being “trapped” at a flat region. That is why we name the function Self-resurrecting Activation Unit.

We contrast SRAU with a “normalized” version, which may seem intuitive because it ensures one pair of gates $B_m^{\text{vis}}[i, j]$ and $B_m^{\text{lan}}[i, j]$ add up to 1.

$$\begin{aligned} [\beta_1; \beta_2] &= \text{SRAU}(\alpha, \tau) \\ \text{NormSRAU}(\alpha, \tau) &= \left[\frac{\beta_1}{\beta_1 + \beta_2}; \frac{\beta_2}{\beta_1 + \beta_2} \right] \end{aligned} \quad (15)$$

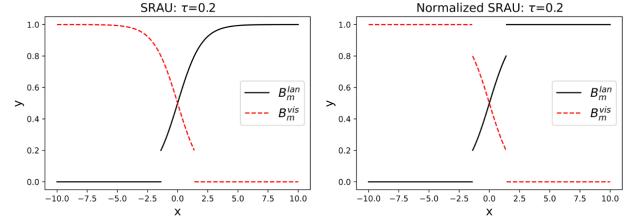


Figure 4. Left: Self-resurrecting activation function with $\tau = 0.2$. Right: Normalized self-resurrecting activation. The x-axis indicates the function inputs and the y-axis indicates function values.

However, the normalization introduces large flat regions of zero gradients, as illustrated in Figure 4 (right). In Section 4.4, we compare the two versions and show the unnormalized SRAU works better.

4. Experiments

4.1. Datasets and Evaluation Metrics

We evaluate our model on the popular MS COCO dataset [35] and the Conceptual Captions dataset [56]. MS COCO contains 123,287 images and each of them is annotated with 5 different captions. We follow the “Karpathy” split [24] for the validation and test set. The Conceptual Captions dataset [56] contains a wider variety of both images and image caption styles than MS COCO. It contains around 3.3M images for training and 28K for validation. As the test data is not publicly available, we instead use the public validation data as our test set, and randomly sample 5000 different image-caption pairs from the training set as the validation set. All the sentences have been converted to lower cases.

To create the small training data setup, we randomly sample 0.1%, 0.5% and 1% image-caption pairs and use them as training data. The procedure is repeated 4 times with different random seeds.

The evaluation metrics include BLEU [45], METEOR [5], ROUGE [34], CIDEr [61] and SPICE [2]. We report the average performance with standard deviation.

4.2. Experimental settings

Baselines. We compare our model with several state-of-the-art transformer-based models, including (1) Plain Transformer [60] model. (2) AoANet [22], which replaces the feed-forward module with an attention-on-attention module in every transformer layer. (3) \mathcal{M}^2 Transformer [12], the current state-of-the-art image-captioning model on MS COCO. As VisualGPT has 12 decoder layers, for fair comparisons, we also create variants of Transformer and \mathcal{M}^2 Transformer with 12-layer decoders.

| Method | Decoder Layers | BLEU-1 | BLEU-4 | METEOR | ROUGE | CIDEr | SPICE |
|----------------------------------|-------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| 0.1% training data | | | | | | | |
| Transformer [60] | 3 | 57.4±0.26 | 13.1±0.15 | 16.7±0.25 | 40.7±0.17 | 40.8±0.15 | 10.3±0.15 |
| \mathcal{M}^2 Transformer [12] | 3 | 56.9±0.14 | 13.1±0.21 | 16.9±0.09 | 40.6±0.24 | 40.9±0.89 | 10.2±0.10 |
| AoANet [22] | 3 | 56.6±1.85 | 13.5±1.81 | 15.9±0.36 | 40.7±1.04 | 38.4±4.30 | 9.9±0.58 |
| Transformer [60] | 12 | 44.0±2.35 | 3.8±0.44 | 9.5±0.40 | 36.0±3.15 | 4.74±1.10 | 2.1±0.15 |
| \mathcal{M}^2 Transformer [12] | 12 | 52.0±1.85 | 9.1±0.31 | 13.7±3.03 | 39.5±1.30 | 33.1±1.00 | 7.8±0.21 |
| AoANet [22] | 12 | 20.7±2.49 | 2.0±2.65 | 7.9±1.56 | 34.0±2.98 | 7.0±1.34 | 3.2±0.49 |
| VisualGPT (Ours) | 12 | 58.2±2.30 | 16.4±0.65 | 18.5±1.85 | 41.9±0.17 | 45.1±1.90 | 10.9±0.40 |
| 0.5% training data | | | | | | | |
| Transformer | 3 | 62.8±0.45 | 18.8±0.64 | 19.4±0.20 | 45.2±0.57 | 59.2±0.41 | 13.0±0.15 |
| \mathcal{M}^2 Transformer | 3 | 63.3±0.10 | 19.4±0.42 | 19.8±0.27 | 45.6±0.12 | 61.3±0.75 | 13.7±0.22 |
| AoANet | 3 | 63.5±1.25 | 20.2±1.56 | 19.4±0.31 | 45.8±0.60 | 63.9±4.30 | 13.8±0.60 |
| Transformer | 12 | 60.9±0.34 | 15.8±0.29 | 18.0±0.12 | 43.1±0.25 | 49.7±1.04 | 11.0±0.08 |
| \mathcal{M}^2 Transformer | 12 | 60.1±1.38 | 14.4±0.10 | 17.9±0.72 | 43.7±0.86 | 44.2±0.30 | 11.4±0.26 |
| AoANet | 12 | 57.9±2.62 | 15.5 ±0.71 | 17.1 ± 0.42 | 43.4± 0.28 | 46.8± 1.13 | 11.3±0.21 |
| VisualGPT (Ours) | 12 | 66.2±1.19 | 22.1±0.96 | 21.1±0.40 | 47.3±0.61 | 70.3±1.70 | 14.6±0.25 |
| 1% training data | | | | | | | |
| Transformer | 3 | 66.0±0.34 | 21.9±0.21 | 21.1±0.15 | 47.3±0.5 | 71.9±1.19 | 14.5±0.15 |
| \mathcal{M}^2 Transformer | 3 | 67.1±0.58 | 23.4±0.42 | 21.3±0.17 | 48.3±0.26 | 73.0±1.0 | 14.9±0.13 |
| AoANet | 3 | 67.6±0.71 | 23.6±1.43 | 21.5±0.31 | 48.4±0.28 | 75.5±2.30 | 15.1±0.32 |
| Transformer | 12 | 64.0±0.56 | 19.6±0.83 | 19.5±0.22 | 45.7±0.49 | 62.1±0.67 | 12.5±0.17 |
| \mathcal{M}^2 Transformer | 12 | 63.3±1.28 | 18.0±0.62 | 19.3±0.64 | 46.1±0.85 | 52.9±5.00 | 12.7±0.55 |
| AoANet | 12 | 63.7 ±0.92 | 17.7 ±0.14 | 18.5 ± 0.14 | 48.2± 2.12 | 58.4± 0.57 | 12.5±0.64 |
| VisualGPT (Ours) | 12 | 69.7±0.62 | 25.7±0.72 | 22.6±0.21 | 49.8±0.21 | 82.5±1.81 | 15.8±0.21 |

Table 1. Testing results by training on small subsets: Performance evaluations of the compared methods training on 0.1%, 0.5% and 1% of MS COCO image-caption pairs. The best performance in each evaluation is indicated in **Bold** font.

4.3. Quantitative Results

Small In-domain Training Data. Results on MS COCO and Conceptual Captions are presented in Tables 1 and 3 respectively. On MS COCO, VisualGPT achieves the best performance among all models. VisualGPT outperforms the best baseline model by 4.2 CIDEr when trained on 0.1% of MS COCO data, 6.4 CIDEr with 0.5% data and 7.0 CIDEr with 1% data. In the experiments on the Conceptual Captions dataset, we compare against only baseline models utilizing 3-layer decoders as these baselines have demonstrate superior performance on MS COCO. Once again, VisualGPT outperforms all the baselines in every matrix evaluation. It outperforms the best baseline model by 4.2 CIDEr under 0.1% training data, 5.4 CIDEr under 0.5% data and 1.4 CIDEr under 1% data.

Comparison against Semi-supervised and unsupervised methods. Kim *et al* [26] proposed a semi-supervised learning method to improve the data efficiency of image captioning. They used 1% of images as training data, rather than 1% of image-caption pairs in Table 1. For Kim *et al* + unpaired, they also employ the other 99% of MS COCO as un-

paired images and captions for training. We replicate their setup. In Table 4, we compare VisualGPT against the results reported in [26]. Without using unpaired images and captions, the proposed VisualGPT method outperforms Kim *et al* by 20.6 CIDEr score.

We also compared VisualGPT against unsupervised methods of Gu *et al* [18] and Feng *et al* [16], which use tens of millions of unpaired images and captions. Even though these are not fair comparisons, it is encouraging to see that only 1133 training images are needed to surpass their performance.

4.4. Ablation Studies

To further quantify the contribution of the pretrained language model and the proposed self-resurrecting encoder-decoder attention, we conduct experiments on the following ablated version of VisualGPT.

- **Base + random init.** This is the base model, a Transformer [60] architecture with a 3-layer encoder, a 12-layer decoder, and a traditional cross-modality attention between the encoder and the decoder. The model

| Ablation | B-1 | B-4 | M | R | C | S |
|----------------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 0.1% training | | | | | | |
| Base + random init. | 44.0 | 3.8 | 9.5 | 36.0 | 4.7 | 2.1 |
| Base + GPT2 init. | 56.8 | 15.3 | 17.0 | 41.2 | 42.9 | 10.5 |
| Base + GPT2 + Meshed | 54.9 | 14.7 | 16.6 | 41.1 | 41 | 10.4 |
| Base + GPT2 + AoA | 55.5 | 14.4 | 16.2 | 40.7 | 40.1 | 10.2 |
| Normalized SRAU | 55.7 | 15.0 | 16.8 | 41.2 | 42.4 | 10.4 |
| Full VisualGPT | 58.2 | 16.4 | 18.5 | 41.9 | 45.1 | 10.9 |
| 0.5% training | | | | | | |
| Base + random init. | 60.9 | 15.8 | 18.0 | 43.1 | 49.7 | 11.0 |
| Base + GPT2 init. | 65.1 | 21.8 | 20.6 | 46.6 | 69.5 | 14.1 |
| Base + GPT2 + Meshed | 64.7 | 21.8 | 20.7 | 47.1 | 68.5 | 14.2 |
| Base + GPT2 + AoA | 64.2 | 21.2 | 20.5 | 46.5 | 67.2 | 13.8 |
| Normalized SRAU | 65.3 | 21.8 | 20.9 | 47.0 | 69.3 | 14.1 |
| Full VisualGPT | 66.2 | 22.1 | 21.1 | 47.3 | 70.3 | 14.6 |
| 1% training | | | | | | |
| Base + random init. | 64.0 | 19.6 | 19.5 | 45.7 | 62.1 | 12.5 |
| Base + GPT2 init. | 68.5 | 25.1 | 22.1 | 49.0 | 80.5 | 15.4 |
| Base + GPT2 + Meshed | 68.2 | 25.0 | 22.4 | 49.2 | 80.4 | 15.4 |
| Base + GPT2 + AoA | 68.5 | 24.6 | 22.0 | 48.6 | 78.4 | 15.0 |
| Normalized SRAU | 69.1 | 25.2 | 22.3 | 49.3 | 81.4 | 15.5 |
| Full VisualGPT | 69.7 | 25.7 | 22.6 | 49.8 | 82.5 | 15.8 |

Table 2. Ablation study on VisualGPT with different encoder-decoder attentions and compare the functionality of the pretrained language model.

| Models | Decoder Layers | B-1 | B-4 | M | R | C |
|-----------------------------|----------------|-------------|------------|------------|-------------|-------------|
| 0.1% training data | | | | | | |
| Transformer | 3 | 12.4 | 2.4 | 4.9 | 15.2 | 21.2 |
| \mathcal{M}^2 Transformer | 3 | 13.1 | 2.8 | 4.8 | 15.5 | 23.5 |
| AoANet | 3 | 11.4 | 2.4 | 4.6 | 14.7 | 20.9 |
| VisualGPT | 12 | 13.9 | 3.2 | 5.6 | 16.7 | 27.7 |
| 0.5% training data | | | | | | |
| Transformer | 3 | 13.2 | 3.3 | 5.5 | 16.3 | 29.6 |
| \mathcal{M}^2 Transformer | 3 | 14.5 | 3.6 | 6.0 | 17.1 | 32.0 |
| AoANet | 3 | 13.8 | 3.3 | 5.6 | 17.9 | 31.8 |
| VisualGPT | 12 | 15.4 | 4.1 | 6.6 | 18.4 | 37.4 |
| 1% training data | | | | | | |
| Transformer | 3 | 13.9 | 3.7 | 6.3 | 18.1 | 37.9 |
| \mathcal{M}^2 Transformer | 3 | 16.0 | 4.1 | 6.8 | 18.9 | 39.8 |
| AoANet | 3 | 14.9 | 4.1 | 6.5 | 18.6 | 39.0 |
| VisualGPT | 12 | 16.4 | 4.3 | 6.9 | 19.2 | 41.2 |

Table 3. Testing results by training on small Conceptual Captions subsets

parameters are randomly initialized instead of pre-trained.

- **Base + GPT2 init.** On top of the base model, we load the GPT-2 pretrained weights into the decoder. Other weights remain randomly initialized.

| Method | B-1 | B-4 | M | R | C |
|------------------------------|-------------|-------------|-------------|-------------|-------------|
| Kim <i>et al.</i> [26] | 58.1 | 13.4 | 15.9 | - | 36.0 |
| Kim <i>et al.</i> + unpaired | 63.0 | 18.7 | 20.7 | - | 55.2 |
| VisualGPT | 67.1 | 24.3 | 21.9 | 48.6 | 75.8 |
| Gu <i>et al.</i> [18] | 46.2 | 5.4 | 13.2 | - | 17.7 |
| Feng <i>et al.</i> [16] | 58.9 | 18.6 | 17.9 | - | 54.9 |

Table 4. Comparison using Kim *et al.*'s split of MS COCO. Kim *et al.* employ only 1% images for training, whereas Kim *et al.* + unpaired also use the rest of training data as unpaired images and texts. We also include unsupervised baselines of Gu *et al.* and Feng *et al.*.

- **Base + GPT2 + Meshed** [12]. On top of the Base + GPT2 init. model, we apply the meshed cross-connection between the encoder and the decoder [12] instead of the traditional cross-modality attention.
- **Base + GPT2 + AoA** [22]. On top of the Base + GPT2 init. model, we add Attention on Attention [22] to the simple cross-modality attention in the decoder.
- **Normalized SRAU.** We replace the self-resurrecting activation unit in VisualGPT with the normalized self-resurrecting activation unit (see Figure 4). We experimented with other activation functions that do not suffer from zero gradients, such as Leaky ReLU and GELU, but the training crashed as the activation values became too large.

Effects of GPT-2 pretrained weights. Comparing the random initialization (Base + random init.) and the GPT-2 pretrained weights (Base + GPT2 init.), it is evident that the GPT-2 weights play a significant role in learning from small data. In particular, the gap between these two models is the most pronounced when training on the least data.

Effects of the proposed encoder-decoder attention. We compare the full VisualGPT model with two other variants of the encoder-decoder attention, Base + GPT2 + Meshed and Base + GPT2 + AOA. The VisualGPT model achieves the best performance in all three setups, demonstrating the effectiveness of the proposed mechanism.

Effects of self-resurrecting activation. In the Normalized SRAU ablation baseline, the self-resurrecting capability of SRAU is eliminated. This results in substantially lowered performance, decreasing CIDEr from Full VisualGPT by 2.7, 1.0, and 0.3 respectively on the three setups. This demonstrates that the self-resurrecting property is beneficial for learning from small data.

4.5. Human Study

We conducted a Amazon Mechanical Turk study to investigate human preferences over the generated captions. We randomly select 50 test images from the three setup

GT: the lady is sitting on the wood bench

| Ours | a | woman | sitting | on | a | bench | in | a | park |
|-----------|-----|-------|---------|------|-----|-------|-----|------|------|
| attention | 0.7 | 0.78 | 0.82 | 0.76 | 0.8 | 0.96 | 0.8 | 0.69 | 0.85 |

GT: a laptop with a keyboard and mouse are on this desk

| Ours | a | laptop | sitting | on | a | desk | with | a | mouse |
|-----------|-----|--------|---------|-----|-----|------|------|------|-------|
| attention | 0.7 | 0.78 | 0.81 | 0.7 | 0.7 | 0.92 | 0.85 | 0.64 | 0.76 |

GT: a cat is sitting in front of a television

| Ours | a | cat | is | sitting | in | front | of | a | television |
|-----------|-----|------|-----|---------|-----|-------|-----|------|------------|
| attention | 0.8 | 0.86 | 0.8 | 0.83 | 0.7 | 0.72 | 0.6 | 0.71 | 0.93 |

GT: a number of people sitting on a snowy surface with skis

| Ours | a | couple | of | people | sitting | on | a | snowy | surface |
|-----------|-----|--------|------|--------|---------|------|------|-------|---------|
| attention | 0.8 | 0.87 | 0.71 | 0.85 | 0.91 | 0.76 | 0.71 | 0.94 | 0.95 |

Figure 5. Visual scores of words in generated captions. We show the raw visual scores and highlight them according to normalized visual scores. High visual scores are in blue and low scores in red.

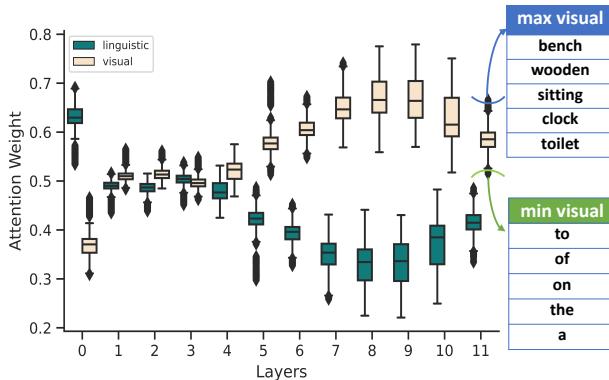


Figure 6. Distributions of linguistic attention (B^{lan}) and visual attention (B^{vis}) at every decoding layer. We also show the words generated with the highest and lowest visual attention.

of 0.1%, 0.5%, and 1% training data. For every image, we generate one caption from VisualGPT and each of three high-performing baselines from Table 1, Transformer [60], \mathcal{M}^2 Transformer [12], and AoANet [22], all with three decoder layers. Every image is evaluated by 5 different Turkers and they need to choose the caption which can most accurately describe the image content. Finally we received 750 valid responses and the results are shown in Table 5.

Overall, we can observe that the captions generated by VisualGPT have received the most votes, 39.6% for the 0.1% split, 38.0% for the 0.5% split, 36.4% for the 1% split. For each training setup, we conducted Pearson’s Chi-square test [46], which shows the differences are statistical significant with $p < 0.05$ in all cases.

| Method | 0.1% data | 0.5% data | 1% data |
|----------------------------------|-----------|-----------|---------|
| Transformer [60] | 19.6% | 19.2% | 17.2% |
| AoANet [22] | 9.6% | 19.2% | 24.4% |
| \mathcal{M}^2 Transformer [12] | 31.2% | 23.6% | 22.0% |
| VisualGPT | 39.6% | 38.0% | 36.4% |

Table 5. The percentage of votes for our VisualGPT and baseline models in 0.1%, 0.5% and 1% training data.

4.6. Qualitative Analysis

In this section, we examine examples from the VisualGPT model trained on 1% of MS COCO. First, we show example captions generated by VisualGPT in Figure 5 and the associated B^{vis} at the last decoder layer. Note that for every word generated, we have a 768-dimensional visual gate vector, which is a slice of B^{vis} at different time steps. We take the mean of the gate vector as the visual score for that word. After that, we normalize the visual scores across the dataset to the $[0, 1]$ interval and highlight the words accordingly. Blue indicates high visual scores and red indicates low visual scores. We observe that, in agreement with our intuition, VisualGPT assigns high visual scores to words like “desk” and “snowy surface” and low visual scores to determiners and prepositions.

In Figure 6, we plot the distribution of B^{vis} and B^{lan} at every decoder layer as a box-and-whisker diagram. We also show the words with the highest and lowest visual scores, which are again in line with our expectations. Additionally, we observe that, going from layer 0 to layer 9, the decoder makes increasing use of visual information, but the uppermost layers, 10 and 11, make more balanced use of information. We hypothesize that the low layers focus on low-level linguistics like syntax, whereas the middle layers learn to fuse linguistic information with visual information. Finally, the two information sources become balanced in the uppermost layers.

5. Conclusion

In this paper, we presented a data efficient image captioning model, VisualGPT, which leverages the linguistic knowledge from the pretrained language model. To bridge the semantic gap between different modalities, we designed a novel encoder-decoder attention mechanism with an unsaturated rectified gating function. We evaluate our model on 0.1%, 0.5% and 1.0% of the MSCOCO dataset. The experimental results demonstrate the effectiveness of our approach, which outperforms several strong baseline models.

References

- [1] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. nocaps: novel object caption-

- ing at scale. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8948–8957, 2019. 1
- [2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016. 5
- [3] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018. 2, 12
- [4] Jyoti Aneja, Aditya Deshpande, and Alexander G Schwing. Convolutional image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5561–5570, 2018. 2
- [5] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. 5
- [6] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003. 3
- [7] Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Aubrey Tatarowicz, Brandy White, Samuel White, and Tom Yeh. Vizwiz: Nearly real-time answers to visual questions. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*, 2010. 1
- [8] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 2, 3
- [9] Paweł Budzianowski and Ivan Vulic. Hello, it’s GPT-2 - how can I help you? towards the use of pretrained language models for task-oriented dialogue systems. In Alexandra Birch, Andrew M. Finch, Hiroaki Hayashi, Ioannis Konstas, Thang Luong, Graham Neubig, Yusuke Oda, and Katsuhiro Sudoh, editors, *Proceedings of the 3rd Workshop on Neural Generation and Translation@EMNLP-IJCNLP 2019, Hong Kong, November 4, 2019*, pages 15–22. Association for Computational Linguistics, 2019. 2
- [10] Aurelia Bustos, Antonio Pertusa, Jose-Maria Salinas, and Maria de la Iglesia-Vayá. Padchest: A large chest x-ray image dataset with multi-label annotated reports. *Medical image analysis*, 66, 2020. 1
- [11] Prithvijit Chattopadhyay, Deshraj Yadav, Viraj Prabhu, Arjun Chandrasekaran, Abhishek Das, Stefan Lee, Dhruv Batra, and Devi Parikh. Evaluating visual conversational agents via cooperative human-ai games. In *HCOMP*, 2017. 1
- [12] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10578–10587, 2020. 1, 2, 5, 6, 7, 8, 12, 13
- [13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. 2, 3
- [14] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015. 2
- [15] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13063–13075, 2019. 2
- [16] Yang Feng, Lin Ma, Wei Liu, and Jiebo Luo. Unsupervised image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4125–4134, 2019. 2, 6, 7
- [17] Sergey Golovanov, Rauf Kurbanov, Sergey Nikolenko, Kyryl Truskovskyi, Alexander Tselousov, and Thomas Wolf. Large-scale transfer learning for natural language generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6053–6058, 2019. 2
- [18] Juxiang Gu, Shafiq Joty, Jianfei Cai, and Gang Wang. Unpaired image captioning by language pivoting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 503–519, 2018. 6, 7
- [19] 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. 12
- [20] Lisa Anne Hendricks, Subhashini Venugopalan, Marcus Rohrbach, Raymond Mooney, Kate Saenko, and Trevor Darrell. Deep compositional captioning: Describing novel object categories without paired training data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–10, 2016. 1
- [21] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 3
- [22] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4634–4643, 2019. 1, 2, 5, 6, 7, 8, 13
- [23] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4565–4574, 2016. 2
- [24] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 1, 5
- [25] Dong-Jin Kim, Jinsoo Choi, Tae-Hyun Oh, and In So Kweon. Image captioning with very scarce supervised data:

- Adversarial semi-supervised learning approach. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2012–2023. Association for Computational Linguistics, 2019. 2
- [26] Dong-Jin Kim, Jinsoo Choi, Tae-Hyun Oh, and In So Kweon. Image captioning with very scarce supervised data: Adversarial semi-supervised learning approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2012–2023, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. 6, 7
- [27] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 12
- [28] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2891–2903, 2013. 1
- [29] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2019. 3
- [30] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2
- [31] Guang Li, Linchao Zhu, Ping Liu, and Yi Yang. Entangled transformer for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8928–8937, 2019. 2
- [32] Xiuju 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. 3
- [33] Yuan Li, Xiaodan Liang, Zhiting Hu, and Eric P Xing. Hybrid retrieval-generation reinforced agent for medical image report generation. In *Advances in Neural Information Processing Systems 31*, pages 1530–1540. 2018. 1
- [34] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 5
- [35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 5, 12
- [36] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved image captioning via policy gradient optimization of spider. In *Proceedings of the IEEE international conference on computer vision*, pages 873–881, 2017. 2
- [37] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv Preprint*, arXiv 1907.11692, 2019. 2
- [38] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. 2019. 3
- [39] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018. 12
- [40] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 13–23, 2019. 3
- [41] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 375–383, 2017. 2
- [42] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7219–7228, 2018. 2
- [43] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. 2017. 2
- [44] Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5528–5531. IEEE, 2011. 3
- [45] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002. 5
- [46] Karl Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900. 8
- [47] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer.

- Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237, 2018. 3
- [48] Di Qi, Lin Su, Jia Song, Edward Cui, Taroon Bharti, and Arun Sacheti. Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data. *arXiv preprint arXiv:2001.07966*, 2020. 3
- [49] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2, 3
- [50] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2, 3
- [51] 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. 2
- [52] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 12
- [53] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024, 2017. 2
- [54] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024, 2017. 12
- [55] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. 3, 12
- [56] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018. 1, 5
- [57] Richard Socher and Li Fei-Fei. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 966–973. IEEE, 2010. 2
- [58] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. ViL-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2020. 3
- [59] Hao Tan and Mohit Bansal. LXMERT: learning cross-modality encoder representations from transformers. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5099–5110. Association for Computational Linguistics, 2019. 3
- [60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3, 5, 6, 8, 13
- [61] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. 5
- [62] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663, 2016. 2
- [63] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015. 1, 2
- [64] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10685–10694, 2019. 2
- [65] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763, 2019. 2
- [66] Benjamin Z Yao, Xiong Yang, Liang Lin, Mun Wai Lee, and Song-Chun Zhu. I2t: Image parsing to text description. *Proceedings of the IEEE*, 98(8):1485–1508, 2010. 2
- [67] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 684–699, 2018. 2

A. Supplementary material

In supplementary material, we provide experimental details and additional qualitative examples.

A.1. Additional implementation details

Image and Word Features. Following [3], we use a Faster R-CNN networks [52] with ResNet-101 [19] as a backbone to train on Visual Genome dataset [27], and we extract a 2048-dimensional feature vector for each object.

We use the Byte Pair Encoding (BPE) [55], which effectively incorporate sub-word information and is beneficial for dealing with out-of-vocabulary words. We employ learnable positional encoding and initialize token embedding from pretrained weights of GPT-2.

Architecture and Hyperparameters. We have 3 layers in the encoder and 12 layers in the decoder with 12 heads in each layer. The hidden size D in each layer is 768. We load the GPT-2 (small) pretrained weights, which has 117M parameters into the decoder. We use the learning rate of $1e^{-4}$ under XE loss and $1e^{-5}$ during the reinforcement learning. We train the models with the AdamW optimizer [39] and a batch size 25. The beam size is equal to 5. The threshold τ is tuned on the validation set for different training data.

A.2. Training Details

We train all the models in two steps. We first train the models with cross-entropy (XE) loss and then finetune them using reinforcement learning. The cross-entropy loss \mathcal{L}_{XE} is the traditional autoregressive classification loss

$$\mathcal{L}_{XE} = - \sum_{t=1}^T \log((w_t | w_{1:t-1})) \quad (16)$$

where $w_{1:T}$ represents the target ground truth sequence.

For reinforcement learning, we employ a variant of Self-Critical Sequence training [54]. Following [12], we sample L sentences, $\hat{w}_{1:T}^1, \dots, \hat{w}_{1:T}^L$, with beam search and use the mean reward from the L sentences as the baseline b . The gradient is

$$\nabla_{\theta} \mathcal{L}_{RL}(\theta) = -\frac{1}{k} \sum_{i=1}^L \left((r(\hat{w}_{1:T}^i) - b) \nabla_{\theta} \log p(\hat{w}_{1:T}^i) \right) \quad (17)$$

where $r(\cdot)$ represents the CIDEr-D reward.

A.3. More training dataset

Figure 7 shows other results obtained by training networks on the 5%, 10%, 20%, 50% and 100% data. VisualGPT outperforms other baseline models when we sample $\leq 20\%$ training data, highlighting its effectiveness on low data regimes.

A.4. Ablation study on τ of our SRAU

To evaluate the effect of different τ of our SRAU, we select the τ equals to 0, 0.1, 0.2 and 0.3 and test on the COCO dataset [35]. In Fig. 8 here, we show that $\tau > 0$ can outperform $\tau = 0$ in most cases. Meaning, SRAU is better than soft gating.

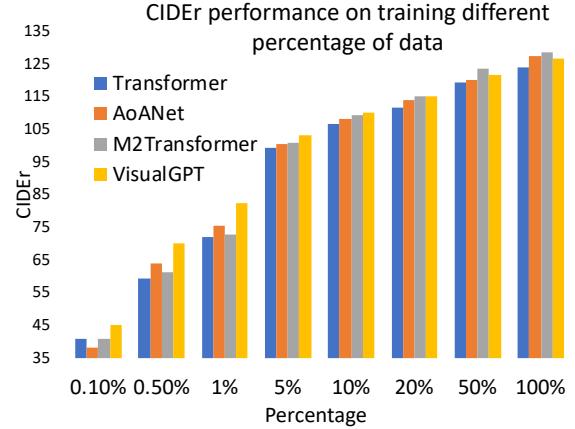


Figure 7. Evaluation on different percentage of data

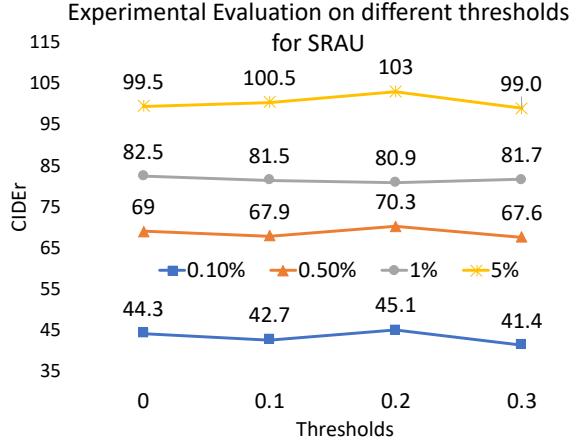


Figure 8. Left: ablation study on different threshold τ .

A.5. Attention over Different types of words

We use the Spacy parser to detect the part-of-speech of words in captions and calculate the mean value of the visual attention score. The result is presented in Fig. 9. We found PoS that tend to visual content, like noun (0.71), verb (0.71) and adjective (0.72), have high visual attention scores, whereas linguistic PoS like pronoun (0.53), punctuation (0.58), and determiner (0.61) receive low attention.

A.6. Hallucination Effect of GPT-2

Directly applying a pretrained language model could potentially suffer from the hallucination and generate words that do not correspond to the image but conform to the a priori knowledge of GPT-2. To evaluate such hallucination effect, we performed a human study on models trained using 1% COCO data. We randomly sample 250 images with the generated caption from each model. For each image, we asked 5 different participants if the caption describes objects not in the image or misses objects in the image (shown in Tables 6 and 7). To catch random clickers, we create 5

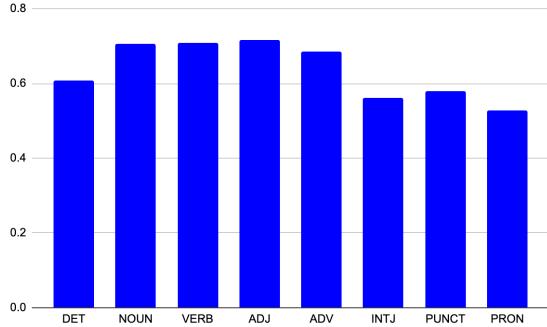


Figure 9. Attention Scores over different part-of-speech words

images with verified captions, and we asked the same questions to 100 participants for each image. Participants who answered these questions wrongly are considered unreliable and removed from the results. Compared to the baselines, VisualGPT has less hallucination and higher coverage of objects.

| Answer | Ours | \mathcal{M}^2 Transformer | Transformer | AoANet | GT |
|---------|-------------|-----------------------------|-------------|--------|------|
| No | 719 | 624 | 633 | 621 | 973 |
| Yes | 367 | 438 | 456 | 447 | 73 |
| No Rate | 0.66 | 0.59 | 0.58 | 0.58 | 0.93 |

Table 6. Does the caption miss things that are shown in the image?

| Answer | Ours | \mathcal{M}^2 Transformer | Transformer | AoANet | GT |
|---------|-------------|-----------------------------|-------------|--------|------|
| No | 720 | 692 | 633 | 655 | 448 |
| Yes | 360 | 418 | 423 | 412 | 43 |
| No Rate | 0.67 | 0.62 | 0.60 | 0.61 | 0.96 |

Table 7. Does the caption describe things that aren't in the image?

A.7. More Qualitative Examples

In Figure 10, we provide more examples of visual attentions. Blue indicates high visual scores and red indicates low visual scores. We can observe that VisualGPT assigns higher scores to words like “steam engine”, “elephants”, “horse”, “lush” and “cabinets”, and it assigns low visual scores to determiners and prepositions like “to” and “at”.

We also show some examples of generated captions by our VisualGPT and several strong baseline models including Transformer (3 layers) [60], \mathcal{M}^2 Transformer (3 layers) [12] and AoANet [22] in the Table 8, Table 9 and Table 10. Overall, we can observe that our VisualGPT is able to describe the image content more accurately than the baseline models.

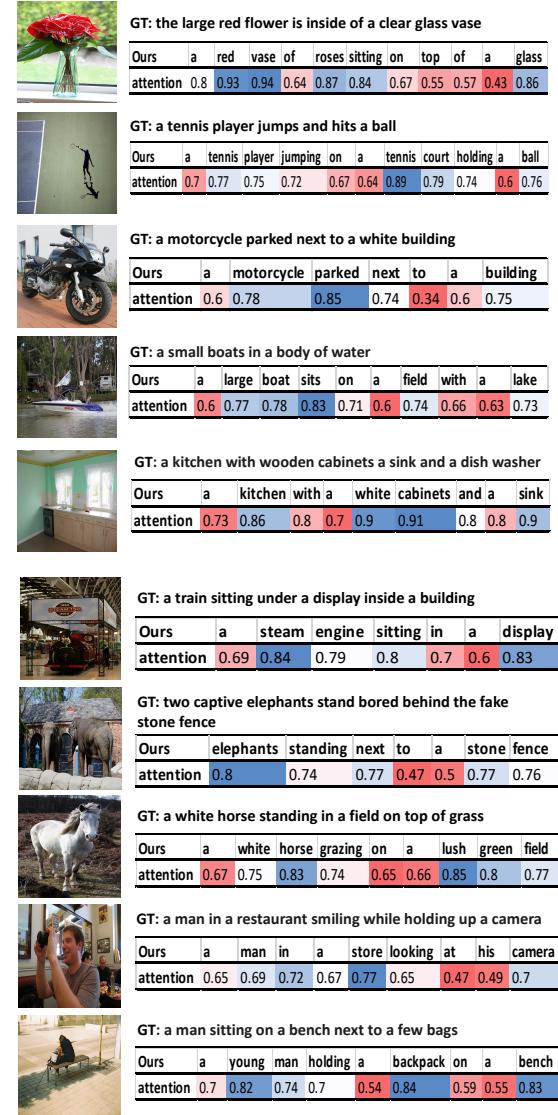


Figure 10. More examples of visual attention for each word in generated captions. High visual scores are in blue and low scores in red.

| Image | Generated Captions | Ground Truth |
|---|--|---|
|  | <p>Transformer: a woman riding some skis on skis \mathcal{M}^2 Transformer: a couple of skiers are standing near the snow AoANet: a man with skis in the snow VisualGPT (ours): a group of people walk on a snowy mountain</p> | <p>GT1: the people are walking through snow in a wooded area GT2: two people wearing skis traveling through the snow GT3: a man is walking down a path covered in a snow GT4: a couple is skiing through the snowy woods GT5: a couple of people that are in a snowy field</p> |
|  | <p>Transformer: a street that has some street in it \mathcal{M}^2 Transformer: a traffic light over a street light under a traffic light AoANet: a street with people on a city street VisualGPT (ours): a street with tall signs and traffic signs</p> | <p>GT1: a yellow traffic light above a street next to houses GT2: a street scene of an intersection with a street light GT3: a stop light hanging over an intersection in a residential area GT4: a traffic signal at an intersection is suspended on wire GT5: a street intersection with a traffic light over it</p> |
|  | <p>Transformer: some pizza are sitting on a plate \mathcal{M}^2 Transformer: a plate with food and a knife on it AoANet: a plate of pizza on a table VisualGPT (ours): a plate of bread are served on a table</p> | <p>GT1: a batch of bread slices sitting on a plate GT2: a plate with some pieces of bread on it GT3: sliced french bread is on a plat that is lying on a table GT4: bread that is sitting on a plate that is on a table GT5: a white plate with lots topped with garlic bread</p> |
|  | <p>Transformer: two tennis player playing tennis on the ball \mathcal{M}^2 Transformer: a tennis player about to hit a ball AoANet: a baseball players on a game playing a game VisualGPT (ours): a tennis player hits a ball with a racket</p> | <p>GT1: a man holding a racquet on top of a tennis court GT2: a man with a tennis racket reaches for a ball GT3: a man with a tennis racket is running on a court GT4: a young man is playing a game of tennis GT5: a tennis player in a blue shirt runs toward a ball</p> |
|  | <p>Transformer: a group of birds that are standing in the grass \mathcal{M}^2 Transformer: a flock of birds perched in a tree branch AoANet: several giraffe are standing next to each trees VisualGPT (ours): a bird standing in the middle of a pond</p> | <p>GT1: a bird is perched a top a branch over a river GT2: a bird sits on a branch above a stream GT3: a bird on top of a tree branch over water GT4: a picture of an outside region that appears incredible GT5: a bird on a fallen branch in a body of water</p> |

Table 8. Caption generated by our VisualGPT, Transformer, \mathcal{M}^2 Transformer and AoANet on 0.1% MS COCO data split

| Image | Generated Captions | Ground Truth |
|---|---|--|
|  | <p>Transformer: several boats are sitting in the middle of a lake</p> <p>\mathcal{M}^2 Transformer: a boat filled with boats floating in the water</p> <p>AoANet: an empty boat that has water and water</p> <p>VisualGPT (ours): a canal filled with boats in the water</p> | <p>GT1: a blue boat docked on a green lush shore</p> <p>GT2: a small marina with boats docked there</p> <p>GT3: a group of boats sitting together with no one around</p> <p>GT4: some boats parked in the water at a dock</p> <p>GT5: boats sitting around the side of a lake by a tree</p> |
|  | <p>Transformer: pizza slices and pizza in a plate covered pizza</p> <p>\mathcal{M}^2 Transformer: people sitting at a table eating pizza and other salad</p> <p>AoANet: two pizza eating a table with pizza on the table</p> <p>VisualGPT (ours): a group of pizza on a iron plate with toppings</p> | <p>GT1: a set of five pizzas sitting next to each other each with different toppings</p> <p>GT2: a handful of prepared pizzas sit next to each other</p> <p>GT3: five uncooked pizzas with a variety of different toppings</p> <p>GT4: five unbaked pizzas that include various types of cheeses</p> <p>GT5: five different pizzas are being prepared over a metal tray</p> |
|  | <p>Transformer: several boats are sitting in the middle of a lake</p> <p>\mathcal{M}^2 Transformer: a boat filled with boats floating in the water</p> <p>AoANet: an empty boat that has water and water</p> <p>VisualGPT (ours): a canal filled with boats in the water</p> | <p>GT1: a blue boat docked on a green lush shore</p> <p>GT2: a small marina with boats docked there</p> <p>GT3: a group of boats sitting together with no one around</p> <p>GT4: some boats parked in the water at a dock</p> <p>GT5: boats sitting around the side of a lake by a tree</p> |
|  | <p>Transformer: a group of people taking a child in a in a building</p> <p>\mathcal{M}^2 Transformer: a group of people in an airport with their hands</p> <p>AoANet: a picture of a young group of people standing for men</p> <p>VisualGPT (ours): a group of people standing around a tv</p> | <p>GT1: a group of men standing around a room</p> <p>GT2: some people are waiting in a long room</p> <p>GT3: people are standing in a room looking at a television screen</p> <p>GT4: a person sitting on a bench while the rest look somewhere else</p> <p>GT5: a man in red winter clothes sits on a bench with people behind him gather in front of a tv</p> |
|  | <p>Transformer: an elephant eating a elephant has a elephant</p> <p>\mathcal{M}^2 Transformer: elephant with its trunk with their elephant with its trunk</p> <p>AoANet: two elephants standing at a lot of trees</p> <p>VisualGPT (ours): three elephants standing next to some trees</p> | <p>GT1: two adult elephants are surrounding a baby elephant</p> <p>GT2: a baby elephant kneeling in front of two bigger elephants</p> <p>GT3: a baby elephant and it 's parents eat fruit</p> <p>GT4: elephants eat fruit a baby elephant rummaging in the food</p> <p>GT5: a pair of adult elephants with a baby elephant eat from a pile of fruit</p> |

Table 9. Caption generated by our VisualGPT, Transformer, \mathcal{M}^2 Transformer and AoANet on 0.5% MS COCO data split

| Image | Generated Captions | Ground Truth |
|---|---|--|
|  | <p>Transformer: a man in a suit and a woman standing in a shop</p> <p>\mathcal{M}^2 Transformer: a man is standing in a shop with a people holding people</p> <p>AoANet: a man is working on a bus in a</p> <p>VisualGPT (ours): a group of people standing at an airport with their luggage</p> | <p>GT1: several people are purchasing tickets at a bus station</p> <p>GT2: some people are checking in at the ticket counter somewhere in asia</p> <p>GT3: people waiting in line with luggage at a ticket counter</p> <p>GT4: people are standing near an airport ticket kiosk</p> <p>GT5: customers stand at a kiosk waiting for tickets</p> |
|  | <p>Transformer: a bus that is parked in front of a building</p> <p>\mathcal{M}^2 Transformer: a couple of people walking down the side of a street</p> <p>AoANet: a bus is parked in a city street</p> <p>VisualGPT (ours): a white and blue bus is parked on the side of a city street</p> | <p>GT1: people standing outside of a blue and white bus</p> <p>GT2: an image of a tour bus that is picking people up</p> <p>GT3: several people standing around buses and most wearing orange vests</p> <p>GT4: a public transit bus pulling up to pick up passengers</p> <p>GT5: a city bus at a stop waiting to pick up passengers</p> |
|  | <p>Transformer: a blue and white airplane flying through a sky</p> <p>\mathcal{M}^2 Transformer: an air plane flying in the air</p> <p>AoANet: a plane airplane flying down in the sky</p> <p>VisualGPT (ours): a plane is flying in the air over the trees</p> | <p>GT1: there 's and airplane in the sky flying over some trees</p> <p>GT2: a large plane is flying over a crowd of trees</p> <p>GT3: a aeroplane soaring high in the sky above the trees</p> <p>GT4: a passenger plane flies in the sky over a forest</p> <p>GT5: an airplane is seen flying over several trees</p> |
|  | <p>Transformer: a white toilet sitting in a white bathroom next to a sink</p> <p>\mathcal{M}^2 Transformer: a cat sitting in the toilet</p> <p>AoANet: a bathroom with a toilet and a sink</p> <p>VisualGPT (ours): a cat sitting on top of a bathroom sink</p> | <p>GT1: a cat climbing into a bathroom sink looking at someone</p> <p>GT2: a cat looks up as it stands in the bathroom sink</p> <p>GT3: a large cat stands inside of a clean bathroom sink</p> <p>GT4: cat is caught stepping in to the bathroom sink</p> <p>GT5: a cute kitty cat in the sink of a bathroom near a brush and other items</p> |
|  | <p>Transformer: a little girl is eating a birthday cake</p> <p>\mathcal{M}^2 Transformer: a child and a child are sitting at a table with table with table</p> <p>AoANet: two children sitting at a table with a laptop computer</p> <p>VisualGPT (ours): a woman and a girl sitting at a table with a birthday cake</p> | <p>GT1: a woman and child stand next to a table with cake on it</p> <p>GT2: a lady standing near the table with a baby is posing for the camera</p> <p>GT3: a woman stands beside a baby in a high chair a table is set with a birthday cake and champagne</p> <p>GT4: a woman setting up her house for a party</p> <p>GT5: a person standing next to a child in a booster seat</p> |

Table 10. Caption generated by our VisualGPT, Transformer, \mathcal{M}^2 Transformer and AoANet on 1% MS COCO data split