AI Communications 35 (2022) 111–129
DOI 10.3233/AIC-210172
IOS Press

Explaining transformer-based image captioning models: An empirical analysis

Marcella Cornia *, Lorenzo Baraldi and Rita Cucchiara
Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, Italy
E-mails: marcella.cornia@unimore.it, lorenzo.baraldi@unimore.it, rita.cucchiara@unimore.it

Abstract. Image Captioning is the task of translating an input image into a textual description. As such, it connects Vision and Language in a generative fashion, with applications that range from multi-modal search engines to help visually impaired people. Although recent years have witnessed an increase in accuracy in such models, this has also brought increasing complexity and challenges in interpretability and visualization. In this work, we focus on Transformer-based image captioning models and provide qualitative and quantitative tools to increase interpretability and assess the grounding and temporal alignment capabilities of such models. Firstly, we employ attribution methods to visualize what the model concentrates on in the input image, at each step of the generation. Further, we propose metrics to evaluate the temporal alignment between model predictions and attribution scores, which allows measuring the grounding capabilities of the model and spot hallucination flaws. Experiments are conducted on three different Transformer-based architectures, employing both traditional and Vision Transformer-based visual features.

Keywords: Image captioning, transformer, explainability, explainable AI

1. Introduction

In the last few years, the integration of vision and language has witnessed a relevant research effort which has resulted in the development of effective algorithms working at the intersection of the Computer Vision and Natural Language Processing domains, with applications such as image and video captioning [2,14,24,27], multi-modal retrieval [13,18,34,51], visual question answering [63,68,85], and embodied AI [3,29,32]. While the performance of these models is constantly increasing and their pervasivity intensifies, the need of explaining models' predictions and quantifying their grounding capabilities is becoming even more fundamental.

In this work, we focus on the image captioning task, which requires an algorithm to describe an input image in natural language. Recent advancements in the field have brought architectural innovations and growth in model size, at the price of lowering the degree of interpretability of the model [24,36]. As a matter of fact, early captioning approaches were based on single-layer additive attention distributions over input regions, which enabled straightforward interpretability of the behavior of the model with respect to the visual input. State-of-the-art models, on the contrary, are based on the Transformer [71] paradigm, which entails multiple layers and multi-head attention, making interpretability and visualization more complex, as the function that connects visual elements to the language model is now highly non-linear. Moreover, the average model size of recent proposals is constantly increasing [62], adding even more challenges to interpretability.

In this paper, we consider a class of recently proposed captioning models, all based on the integration of a fully-attentive encoder with a fully-attentive decoder, in conjunction with techniques for handling a-priori semantic knowledge and multi-level visual features. The reason behind the choice of this class of model is due to their performances in multi-modal tasks [14,75], which at present seem to perform favorably when compared to other Transformer-based architectural choices [36,83]. We firstly verify and foster the interpretability of such models by developing solutions for visualizing what the model concentrates on in the input image, at each step of the generation. Being

[*] Corresponding author. E-mail: marcella.cornia@unimore.it.

the model a non-linear function, we consider attribution methods based on gradient computation, which allow the generation of an attribution map that highlights portions of the input image on which the model has concentrated while generating each word of the output caption. Furthermore, on the basis of these visualizations, we quantitatively evaluate the temporal alignment between model predictions and attribution scores – a procedure that allows to quantitatively identify pitfalls in the grounding capabilities of the model and to discover hallucination flaws, in addition to measuring the memorization capabilities of the model. To this aim, we propose a new alignment and grounding metric which does not require access to ground-truth data.

Experimental evaluations are conducted on both the COCO dataset [38] for image captioning, and on the ACVR Robotic Vision Challenge dataset [21] which contains simulated data from a domestic robot scenario. In addition to considering different model architectures, we also consider different image encoding strategies: the well-known usage of a pre-trained object detector to identify relevant regions [2] and the emerging Vision Transformer [17]. Experimental results will shed light on the grounding capabilities of state-of-the-art models and on the challenges posed by the adoption of the Vision Transformer model.

To sum up, our main contributions are as follows:

- We consider different encoder-decoder architectures for image captioning and evaluate their interpretability by developing solutions for visualizing what the model concentrates on the input image at each step of the generation.
- We propose an alignment and grounding metric which evaluates the temporal alignment between model predictions and attribution scores, thus identifying defects in the grounding capabilities of the model.
- We conduct extensive experiments on the COCO dataset and the ACVR Robotic Vision Challenge dataset, considering different model architectures, visual encoding strategies, and attribution methods.

2. Related work

2.1. Image captioning

Before the advent of deep learning, traditional image captioning approaches were based on the generation of simple template sentences, which were later filled by the output of an object detector or an attribute predictor [60,80]. With the surge of deep neural networks, captioning has started to employ RNNs as language models and the output of one or more layers of a CNN was employed to encode visual information and to condition the generation of language [16,26,33,56,73]. On the training side, initial methods were based on a time-wise cross-entropy training. A notable achievement has then been made with the introduction of the REINFORCE algorithm, which enabled the use of non-differentiable caption metrics as optimization objectives [39,54,56]. On the image encoding side, instead, additive attention mechanisms have been adopted to incorporate spatial knowledge, initially from a grid of CNN features [42,77,82], and then using image regions extracted with an object detector [2,43,48]. To further improve the encoding of objects and their relationships, graph convolution neural networks have been employed as well [78,81], to integrate semantic and spatial relationships between objects or to encode scene graphs.

After the emergence of convolutional language models, which have been explored for captioning as well [4], new fully-attentive paradigms [15,65,71] have been proposed and achieved state-of-the-art results in machine translation and language understanding tasks. Likewise, recent approaches have investigated the application of the Transformer model [71] to the image captioning task [8,35,79], also proposing variants or modifications of the self-attention operator [10,14,20,23,24,46]. Transformer-like architectures can also be applied directly on image patches, thus excluding or limiting the usage of the convolutional operator [17,70]. On this line, Liu et al. [40] devised the first convolution-free architecture for image captioning. Specifically, a pre-trained Vision Transformer network (*i.e.* ViT [17]) is adopted as encoder and a standard Transformer decoder is employed to generate captions.

Other works using self-attention to encode visual features achieved remarkable performance also thanks to vision-and-language pre-training [41,68] and early-fusion strategies [36,85]. For example, following the BERT architecture [15], Zhou et al. [85] devised a single stream of Transformer layers, where region and word tokens are early fused together into a unique flow. This model is first pre-trained on large amounts of image-caption pairs to perform both bidirectional and sequence-to-sequence prediction tasks and then finetuned for image captioning. Li et al. [36]

proposed OSCAR, a BERT-like architecture that also includes objects tags, extracted from an object detector and concatenated with the image regions and word embeddings fed to the model. They also performed a large-scale pre-train with 6.5 million image-text pairs, with a masked token loss similar to the BERT mask language loss and a contrastive loss for distinguishing aligned words-tags-regions triples from polluted ones. On the same line, Zhang et al. [83] proposed VinVL, built on top of OSCAR.

A recent and related research line is that of employing large-scale pre-training for obtaining better image representations, either in a discriminative or generative fashion. For instance, in the CLIP model [51] the pre-training task of predicting which caption goes with which image is employed as an efficient and scalable way to learn state-of-the-art visual representations suitable for multi-modal tasks, while in DALL-E [53] a transformer is trained to autoregressively model text and image tokens as a single stream of data, and achieves competitive performances with previous domain-specific models when evaluated in a zero-shot fashion for image generation.

2.2. *Explainability and visualization in vision-and-language*

While the performance of captioning algorithms has been increasing in the last few years, and while these models are approaching the level of quality required to be run in production, providing effective visualizations of what the model is doing, and explanations to why it fails, is still under-investigated [50,57,59,67]. It shall be noted, in this regard, that early captioning models based on additive attention were easy to be visualized – as their attentive distribution was a single-layer weighted summation of visual features. In the case of modern captioning models, instead, each head of each encoder/decoder layer takes an attentive distribution, thus making visualization less intuitive and straightforward. A solution that is becoming quite popular is that of employing an attribution method, which allows attributing predictions to visual features even in presence of significant non-linearities. For instance, [14,25] used Integrated Gradients [67] to provide region-level attention visualization. Sun et al. [66], instead, develops better visualization maps by employing layer-wise relevance propagation and gradient-based explanation methods, tailored to image captioning models. Following this line, in this work, we visualize attentional states by employing different attribution methods.

A related research line is that of grounded description, which aims at ensuring that objects mentioned in the output captions match with the attentive distribution of the model, thus avoiding hallucination and improving localization [9,44,84]. Some works also integrated saliency information (*i.e.* what do humans pay more attention to in a scene [11,74]) in captioning generation. This idea was first proposed by Sugano and Bulling [64] who exploited human eye-fixation information for image captioning by including normalized fixation histograms over the image as an input to the soft-attention module of [77] and weighing the attended image regions based on whether these are fixated or not. Subsequent works on this line [12,52,69] used predicted saliency information in place of eye-fixation information.

3. Transformer-based image captioning

Regardless of the variety of methodologies and architectures which have been proposed in the last few years, all image captioning models can be logically decomposed into a visual encoder module, in charge of processing visual features, and a language model, in charge of generating the actual caption [62]. In this work we focus on Transformer-based captioning models, hence both the visual encoding step and the language generation step are carried out employing Transformer-like layers and attention operations.

Following the original Transformer architecture [71], multi-modal connections between the two modules are based on the application of the cross-attention operator, thus leaving self-attentive connections between modalities separated from each other. An alternative to this choice which is worth mentioning is that of employing a joint self-attentive encoding on both modalities (*e.g.*, by concatenating visual features and words, and feeding the result to a Transformer encoder) [36,83]. While this early-fusion strategy has become particularly popular in the context of Vision-and-Language pre-training [76], as it allows the application of contrastive-like pre-training losses, we leave it apart in favor of a separate encoding of the modalities as this is still the dominant approach in image captioning when pre-training is not employed [14,24]. Further, a separate encoding of the two modalities has been demonstrated to perform favorably also when large-scale pre-training is employed [75]. In this choice, we are further motivated by preliminary experiments that suggest that an early-fusion strategy does not bring significant performance improvements when used in a fully supervised training setting.

3.1. Image encoding

Following recent advancements in the field, we consider two image encoding approaches: that of extracting regions from the input image through an object detector pre-trained to detect objects and attributes, and that of employing grid features extracted from a Vision Transformer model [17]. While the first approach has become the default choice in the last few years after the emergence of the Up-Down model [2], the Vision Transformer [17] is quickly becoming a compelling alternative, which we explore in this paper. To enrich the quality of the visual features, we employ a Vision Transformer model which is pre-trained for multi-modal retrieval [51]. To the best of our knowledge, the application of Vision Transformers in an image captioning model has been investigated by Liu et al. [40], using the original model trained for image classification, and by Shen et al. [58], which adopted a pre-training based on multi-modal retrieval as well.

Image region features. In this approach, Faster R-CNN [55] is adopted to detect objects in the input image in two stages: the first, called Region Proposal Network, produces object proposals rolling over intermediate features of a CNN; the second operates a pooling of the region of interest to extract a feature vector for each proposal. One of the key elements of this approach resides in its pre-training strategy, where an auxiliary training loss is added for learning to predict attribute classes alongside object classes on the Visual Genome dataset [30]. This allows the model to predict a dense and rich set of detections, including both salient object and contextual regions and favors the learning of better feature representations. In particular, we employ a Faster R-CNN [55] model with a ResNet-101 [22] backbone, finetuned on the Visual Genome dataset [30]. This feature extraction process results in a variable number of detections for an input image, each described by a 2048-d feature descriptor.

Vision transformer (ViT). Transformer-like architectures can be applied directly on image patches, thus excluding or limiting the usage of the convolutional operator [17,70]. The image encoder we employ in this case is a ViT-B/32 which closely follows the original implementation [17] with only the minor modification of adding an additional layer normalization to the combined patch and position embeddings before the transformer and use a slightly different initialization scheme. The encoder is pre-trained on with a symmetric cross-entropy loss for multi-modal retrieval, on a private large-scale dataset [51]. In this case, the image encoding process results in a set of 50 768-d feature vectors, corresponding to either one of the 49 32 × 32 patches in which the input image is divided, or to the extra "classification token" of the architecture.

3.2. Transformer layers

Regardless of the choice, both image encoding methodologies result in a set of feature vectors describing the input image. These are then fed to the encoder part of the architecture (Fig. 1). Both encoder and decoder consist of a stack

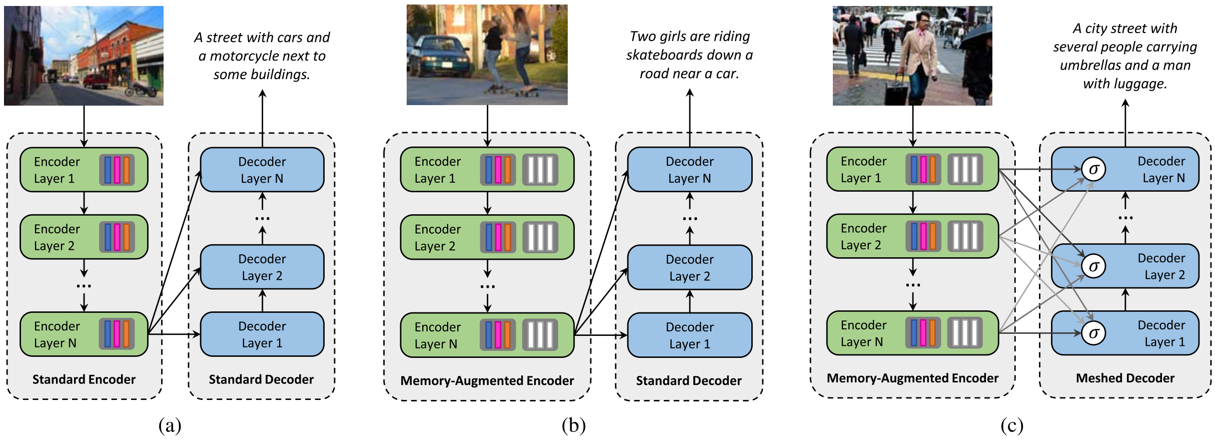


Fig. 1. Structure of the image captioning models considered in our analysis: (a) a standard fully-attentive encoder-decoder captioner; (b) an encoder with additional memory slots to encode a-priori information; (c) a decoder with meshed connectivity which weights contributions from all encoder layers.

of Transformer layers that act, indeed, on image regions and words. In the following, we revise their fundamental features. Each encoder layer consists of a self-attention and feed-forward layer, while each decoder layer is a stack of one self-attentive and one cross-attentive layer, plus a feed-forward layer. Both attention layers and feed-forward layers are encapsulated into "add-norm" operations, described in the following.

Multi-head attention. The core component of both self-attention and cross-attention layers is an attention mechanism [6] with multiple heads with different learned weights. Attention is applied using scaled dot-products as similarity measure [71] while keys, queries, and values are computed through linear transformations.

Formally, given two sequences $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ and $\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \dots, \hat{\mathbf{x}}_M$ of d-dimensional input vectors, each head applies two linear transformations to the first sequence to form key and value vectors:

$$\mathbf{k}_t = \mathbf{W}_k \mathbf{x}_t, \qquad \mathbf{v}_t = \mathbf{W}_v \mathbf{x}_t, \tag{1}$$

where $\mathbf{W}_k$ and $\mathbf{W}_v$ are the key and value transformation matrices, with size $d_h \times d$, where $d_h = d/H$ is the dimensionality of a single head, and H is the number of heads. Analogously, a linear transformation is applied to the second sequence to obtain query vectors:

$$\mathbf{q}_t = \mathbf{W}_q \hat{\mathbf{x}}_t, \tag{2}$$

where $\mathbf{W}_q$ has the same size of $\mathbf{W}_k$ and $\mathbf{W}_v$. Query vectors are used to compute a similarity score with key vectors and generate a weight distribution over values. The similarity score between a query $\mathbf{q}_t$ and a key $\mathbf{k}_c$ is computed as a scaled dot-product between the two vectors, *i.e.* $(\mathbf{q}_t^\intercal \mathbf{v}_c)/\sqrt{d_h}$. Each head then produces a vector by averaging the values $\{\mathbf{v}_c\}_c$ with the weights defined by an attentive distribution over the similarity scores:

$$\mathbf{y}_t = \sum_{c=1}^{N} \alpha_{tc} \mathbf{v}_c, \quad \text{where} \tag{3}$$

$$\alpha_{tc} = \frac{\exp(\mathbf{q}_t^\intercal \mathbf{v}_c/\sqrt{d_h})}{\sum_i \exp(\mathbf{q}_t^\intercal \mathbf{v}_i/\sqrt{d_h})}. \tag{4}$$

Results from different heads are then concatenated and projected to a vector with dimensionality d through a final linear transformation.

In the encoding stage, the sequence of visual features is used to infer queries, keys, and values, thus creating a self-attention pattern in which pairwise visual relationships are modeled. In the decoder, we instead apply both a cross-attention and a masked self-attention pattern. In the former, the sequence of words is used to infer queries, and visual elements are used as keys and values. In the latter, the left-hand side of the textual sequence is used to generate keys and values for each element of the sequence, thus enforcing causality in the generation.

Position-wise feed-forward layers. The second component of a Transformer layer is a fully-connected forward layer which is applied time-wise over the input sequence. This consists of two affine transformations with a single non-linearity,

$$\texttt{FF}(\mathbf{x}_t) = \mathbf{U}\sigma(\mathbf{V}\mathbf{x}_t + \mathbf{b}) + \mathbf{c}, \tag{5}$$

where $\sigma(x) = \max(x, 0)$ is the ReLU activation function, and $\mathbf{V}$ and $\mathbf{U}$ are learnable weight matrices, respectively with sizes $d \times d_f$ and $d_f \times d$; $\mathbf{b}$ and $\mathbf{c}$ are bias terms. The size of the hidden layer d_f is usually chosen to be larger than d, *e.g.* four times d in most implementations [71].

Skip connection and layer normalization. Each sublayer (attention or position-wise feed-forward) is encapsulated within a residual connection [22] and layer normalization [5]. This "add-norm" operation is defined as

$$\texttt{AddNorm}(\mathbf{x}_t) = \texttt{LayerNorm}\big(\mathbf{x}_t + f(\mathbf{x}_t)\big), \tag{6}$$

where f indicates either an attention layer or a position-wise feed-forward layer.

3.3. Fully-attentive encoder

Given a set of visual features extracted from an input image, attention can be used to obtain a permutation invariant encoding through self-attention operations. As noted in [10,14], attentive weights depend solely on the pairwise similarities between linear projections of the input set itself. Therefore, a sequence of Transformer layers can naturally encode the pairwise relationships between visual features. Because everything depends solely on pairwise similarities, though, self-attention cannot model a priori knowledge on relationships between visual features. For example, given a visual feature encoding a glass of fruit juice and one encoding a plate with a croissant, it would be difficult to infer the concept of *breakfast*, which could, instead, be easily inferred using a priori knowledge on relationships.

Memory-augmented attention. In this paper, we consider both an encoder made of original Transformer layers [71], and the memory-augmented encoder proposed in [10,14], which overcomes the aforementioned limitation. In the latter case, the set of keys and values used for self-attention is extended with additional "slots" which can encode a priori information – and which are implemented as plain learnable vectors which can be directly updated via SGD. Formally, given the set of visual features $\mathbf{X}$ expressed in matrix form, the operator is defined as:

$$\begin{aligned}\tilde{\mathbf{X}} &= \mathsf{Attention}(W_q\mathbf{X}, \mathbf{K}, \mathbf{V}) \\ \mathbf{K} &= [W_k\mathbf{X}, \mathbf{M}_k] \\ \mathbf{V} &= [W_v\mathbf{X}, \mathbf{M}_v],\end{aligned} \tag{7}$$

where $\mathsf{Attention}$ indicates the classic attention operator, $\mathbf{M}_k$ and $\mathbf{M}_v$ are learnable matrices with n_m rows, and $[\cdot, \cdot]$ indicates concatenation. Intuitively, by adding learnable keys and values, through attention it will be possible to retrieve learned knowledge which is not already embedded in $\mathbf{X}$.

Just like the self-attention operator, memory-augmented attention can be applied in a multi-head fashion. In this case, the memory-augmented attention operation is repeated h times, using different projection matrices W_q, W_k, W_v, and different learnable memory slots $\mathbf{M}_k$, $\mathbf{M}_v$ for each head. Then, we concatenate the results from different heads and apply a linear projection.

Full encoder. Multiple encoding layers are stacked in sequence so that the i-th layer consumes the output set computed by layer $i-1$. This amounts to creating multi-level encodings of the relationships between visual features, in which higher encoding layers can exploit and refine relationships already identified by previous layers, eventually using a priori knowledge, in the case of the memory-augmented attention. A stack of N encoding layers will therefore produce a multi-level output $\tilde{\mathcal{X}} = (\tilde{\mathbf{X}}^1, \dots, \tilde{\mathbf{X}}^N)$, obtained from the outputs of each encoding layer.

3.4. Fully-attentive decoder

The language model we employ is composed of a stack of decoder layers, each performing self-attention and cross-attention operations. As mentioned, each cross-attention layer uses the outputs from the decoder to infer keys and values, while self-attention layers rely exclusively on the input sequence of the decoder. However, keys and values are masked so that each query can only attend to keys obtained from previous words, *i.e.* the set of keys and values for query $\mathbf{q}_t$ are, respectively, $\{\mathbf{k}_i\}_{i\leqslant t}$ and $\{\mathbf{v}_i\}_{i\leqslant t}$. We consider two variants of the decoder: the standard decoder, which follows the implementation of [71], and the meshed decoder, proposed in [14].

Standard decoder. In the standard decoder, only the activations from the last encoding layer, *i.e.* $\tilde{\mathbf{X}}^N$, are employed to perform cross-attention in all decoding layers.

Meshed decoder. In the meshed decoder, we instead employ all the intermediate activations of the encoder, *i.e.* $(\tilde{\mathbf{X}}^1, \dots, \tilde{\mathbf{X}}^N)$. Given an input sequence of word vectors $\mathbf{Y}$, and outputs from all encoding layers $\tilde{\mathcal{X}}$, the Meshed

Attention operator connects $\mathbf{Y}$ to all elements in $\tilde{\mathcal{X}}$ through gated cross-attentions. These multi-level contributions are then summed together after being modulated. Formally, the operator is defined as

$$\mathsf{MeshedAttention}(\tilde{\mathcal{X}}, \mathbf{Y}) = \sum_{i=1}^{N} \boldsymbol{\alpha}_i \odot \mathsf{CrossAttention}\big(\tilde{\mathbf{X}}^i, \mathbf{Y}\big), \tag{8}$$

where $\mathsf{CrossAttention}(\cdot, \cdot)$ stands for the encoder-decoder cross-attention, computed using queries from the decoder and keys and values from the encoder, and $\boldsymbol{\alpha}_i$ is a matrix of weights having the same size as the cross-attention results. Weights in $\boldsymbol{\alpha}_i$ modulate both the single contribution of each encoding layer, and the relative importance between different layers. These are computed by measuring the relevance between the result of the cross-attention computed with each encoding layer and the input query, as follows:

$$\boldsymbol{\alpha}_i = \sigma\big(W_i\big[\mathbf{Y}, \mathsf{CrossAttention}\big(\tilde{\mathbf{X}}^i, \mathbf{Y}\big)\big] + b_i\big), \tag{9}$$

where $[\cdot, \cdot]$ indicates concatenation, σ is the sigmoid activation, W_i is a $2d \times d$ weight matrix, and b_i is a learnable bias vector.

3.5. Training strategy

At training time, the input of the encoder is the ground-truth sentence $\{\mathrm{BOS}, w_1, w_2, \dots, w_n\}$, and the model is trained with a cross-entropy loss to predict the shifted ground-truth sequence, *i.e.* $\{w_1, w_2, \dots, w_n, \mathrm{EOS}\}$, where BOS and EOS are special tokens to indicate the start and the end of the caption.

While at training time the model jointly predicts all output tokens, the generation process at prediction time is sequential. At each iteration, the model is given as input the partially decoded sequence; it then samples the next input token from its output probability distribution, until a EOS marker is generated.

Following previous works [2,54,56], after a pre-training step using cross-entropy, we further optimize the sequence generation using Reinforcement Learning. Specifically, we employ a variant of the self-critical sequence training approach [56] which applies the REINFORCE algorithm on sequences sampled using Beam Search [2]. Further, we baseline the reward using the mean of the rewards rather than greedy decoding as done in [2,56].

Specifically, given the output of the decoder we sample the top-k words from the decoder probability distribution at each timestep, and always maintain the top-k sequences with highest probability. We then compute the reward of each sentence $\mathbf{w}^i$ and backpropagate with respect to it. The final gradient expression for one sample is thus:

$$\nabla_\theta L(\theta) = -\frac{1}{k}\sum_{i=1}^{k}\big(\big(r\big(\mathbf{w}^i\big) - b\big)\nabla_\theta \log p\big(\mathbf{w}^i\big)\big) \tag{10}$$

where $b = (\sum_i r(\mathbf{w}^i))/k$ is the baseline, computed as the mean of the rewards obtained by the sampled sequences. To reward the overall quality of the generated caption, we use image captioning metrics as a reward. Following previous works [2], we employ the CIDEr metric (specifically, the CIDEr-D score) which has been shown to correlate better with human judgment [72].

4. Providing explanations

Given the predictions by a captioning model, we develop a methodology for explaining what the model attends in the input image and quantitatively measure its grounding capabilities and the degree of alignment between its predictions and the visual inputs it attends. To this aim, we first extract attention visualizations over input visual elements, as a proxy of what the model attends at each timestep during the generation. These, depending on the model, can be either object detections or raw pixels.[1] Being the model a highly non-linear function, as anticipated we resort to the usage of an attribution method [59,61,67] to calculate a score map over each visual element. The score map encodes the contribution of the visual element to each predicted word: by aggregating scores on visual elements over object regions, the grounding and alignment capabilities of the model can be evaluated by considering the intersection between attended regions and predicted words.

4.1. *Attribution methods*

Given a predicted sentence $\{w_0, w_1, \dots, w_T\}$, for each predicted word we compute an attribution function $A_t(I_0)$ that ranks the visual elements of the input image I_0 based on their influence on the score of w_t. On this line, we employ different attribution methods based on gradient computation that are applicable, namely Saliency, Guided Backpropagation, and Integrated Gradients.

Saliency. Originally proposed in [59], it approximates a non-linear captioning model in the neighborhood of I_0 with a first-order Taylor expansion. Formally, it expresses the score for w_t produced by the model as:

$$S_w(I) \approx \left.\frac{\partial S_w}{\partial I}\right|_{I_0} I + b, \tag{11}$$

where b is a bias. Under this approximation, the magnitude of the partial derivative of the output of the model with respect to the input can be used to identify which input elements need to be changed to affect the output the most. The attribution score, in this case, is thus simply given by the value of the partial derivative reported in Eq. (11).

Guided backprop. Guided backpropagation [61] computes the gradient of the target word with respect to the input, as the Saliency approach does. In this case, though, gradients of ReLU functions are overridden so that only non-negative gradients are backpropagated.

Integrated gradients. Integrated Gradients [67] is an axiomatic model that works by approximating the integral of gradients of the model output with respect to the input, along the path from a given baseline to the input itself. Formally, the attribution function is defined as follows in this case:

$$A_t(I_0) = (I_b - I_0) \cdot \int_{\alpha=0}^{1} \left.\frac{\partial S_w}{\partial I}\right|_{I_b+\alpha\cdot(I_b-I_0)} d\alpha, \tag{12}$$

where I_b is a baseline input image, which is our case is a zero tensor.

4.2. *Evaluating grounding capabilities*

Once the attribution map has been computed, we aggregate the scores over visual regions, as computed from a Faster-RCNN model pre-trained on Visual Genome [2]. For the sake of interpretability and to enhance visualization, we normalize the scores between 0 and 1 applying a contrast stretching-like normalization [19]. At each timestep of the caption, we then identify the region with the highest attribution score. To measure the grounding and alignment capabilities of the model, we check whether each noun that has been predicted by the model is semantically similar to a region which the model has attended, either at the same timestep or in previous ones.

Our grounding score, given a temporal margin δ, can be formally defined as follows:

$$\mathsf{GroundingScore}(\delta) = \sum_{t \in \mathcal{N}} \max_{\tau \in [t-\tau, t]} \mathrm{sim}(w_t, c_\tau), \tag{13}$$

[1] During the rest of this section, we will employ the term "visual elements" to interchangeably indicate image pixels or regions.

where sim indicates the GloVe [49] similarity, w_t is the t-th word predicted by the captioner, and c_τ is the semantic class of the region with the highest attribution score at timestep τ. $\mathcal{N}$ is the set of timesteps in which a noun is predicted by the captioner. The grounding score is then averaged over the entire dataset.

As it can be observed, the proposed score increases monotonically when increasing δ. A model with good alignment capabilities, though, will have high score values starting from low values of δ, meaning that it attends correct regions right before producing the corresponding words. A model with low alignment capabilities, but with good memory properties, will instead reach high score values for higher δ values, meaning that it attends and memorizes regions well before generating the corresponding words. On the contrary, a model with low grounding capabilities (or with hallucination flaws) will have low score values on average, regardless of δ.

5. Experimental evaluation

5.1. Datasets

To train and test the considered Transformer-based captioning models, we employ the Microsoft COCO dataset [38], which contains 123 287 images labeled with 5 captions each. We employ the data splits defined in [27], where 5 000 images are used for validation, 5 000 images for testing, and the rest for training. Further, we also validate our approach and the proposed metric on images taken from a robot-centric point of view. To this end, we employ the ACVR Robotic Vision Challenge dataset [21] which contains simulated data from a domestic robot scenario. The dataset contains scenes with cluttered surfaces, and day and night lighting conditions. Authors have simulated domestic service robots of multiple sizes, resulting in sequences with three different camera heights above the ground plane. We employ the validation set of this dataset that consists of over 21 000 images in four simulated indoor video sequences.

5.2. Captioning metrics

We employ popular captioning metrics to evaluate both fluency and semantic correctness: BLEU [47], ROUGE [37], METEOR [7], CIDEr [72], and SPICE [1]. BLEU is a form of precision of word n-grams between predicted and ground-truth sentences. As done in previous works, we evaluate our predictions with BLEU using n-grams of length 1 and 4 (*i.e.* B-1 and B-4). ROUGE (R) computes an F-measure with a recall bias using the longest common sub-sequence technique. METEOR (M), instead, scores captions by aligning them to one or more ground-truths. Alignments are based on exact, stem, synonym, and paraphrase matches between words and phrases. CIDEr (C) computes the average cosine similarity between n-grams found in the generated caption and those found in reference sentences, weighting them using TF-IDF. SPICE (S), finally, considers matching tuples extracted from the candidate and the reference scene graphs, thus favoring the semantic content rather than the fluency of generated captions. While it has been shown experimentally that BLEU and ROUGE have a lower correlation with human judgments than the other metrics [72], the common practice in the image captioning literature is to report all the mentioned metrics. To ensure a fair evaluation, we use the Microsoft COCO evaluation toolkit to compute all scores.

In addition, we also evaluate the capability of the captioning approaches to name objects on the scene. To evaluate the alignment between predicted and ground-truth nouns, we employ an alignment score [9] based on the Needleman-Wunsch algorithm [45]. Specifically, given a predicted caption $\boldsymbol{y}$ and its target counterpart $\boldsymbol{y}^*$, we extract all nouns from both sentences and evaluate the alignment between them. We use the following scoring system: the reward for matching two nouns is equal to the cosine similarity between their word embeddings; a gap gets a negative reward equal to the minimum similarity value, *i.e.* -1. Once the optimal alignment is computed, we normalize its score, $\mathrm{al}(\boldsymbol{y}, \boldsymbol{y}^*)$, with respect to the length of the sequences. The alignment score is thus defined as:

$$\mathrm{NW}\big(\boldsymbol{y}, \boldsymbol{y}^*\big) = \frac{\mathrm{al}(\boldsymbol{y}, \boldsymbol{y}^*)}{\max(\#\boldsymbol{y}, \#\boldsymbol{y}^*)} \tag{14}$$

where $\#\boldsymbol{y}$ and $\#\boldsymbol{y}^*$ represent the number of nouns contained in $\boldsymbol{y}$ and $\boldsymbol{y}^*$, respectively. Notice that $\mathrm{NW}(\cdot, \cdot) \in [-1, 1]$.

To assess instead how the predicted caption covers all the objects without considering the order in which they are named in the caption, we employ a soft coverage measure between the ground-truth set of object classes and the set of names in the caption [10]. In this case, we first compute the optimal assignment between predicted and ground-truth nouns using distances between word vectors and the Hungarian algorithm [31]. We then define an intersection score between the two sets as the sum of assignment profits. The coverage measure is computed as the ratio of the intersection score and the number of ground-truth nouns:

$$\mathrm{Cov}\big(\boldsymbol{y}, \boldsymbol{y}^*\big) = \frac{\mathrm{I}(\boldsymbol{y}, \boldsymbol{y}^*)}{\#\boldsymbol{y}^*}, \tag{15}$$

where $\mathrm{I}(\cdot, \cdot)$ is the intersection score, and the # operator represents the cardinality of the set of nouns.

For both NW and Coverage metrics, we employ GloVe word embeddings [49] to compute the similarity between predicted and ground-truth nouns.

5.3. Implementation details

In our experiments, we train and test three different captioning models. For convenience and for coherency with the original papers in which these approaches have been firstly proposed, we refer to a captioner that follows the structure of the original Transformer [71] with three layers as “Transformer”, to a captioner that employs memory-augmented attention and two layers as “SMArT” [10], and to a captioner that employs both memory-augmented attention and a mesh-like connectivity, with three layers as “$\mathcal{M}^2$ Transformer” [14]. As previously mentioned, we train two versions of each captioning model: one using image region features extracted from Faster R-CNN trained on Visual Genome [2,55] and the other employing the Vision Transformer [17] as feature extractor. For all captioning models, we set the dimensionality d of all layers is to 512 and use a number of heads H equal to 8. The dimensionality of the inner feed-forward layer d_f is 2048. We use dropout with keep probability 0.9 after each attention layer and after position-wise feed-forward layers. Input words are represented with one-hot vectors and then linearly projected to the input dimensionality of the model, d. We also employ sinusoidal positional encodings [71] to represent word positions inside the sequence and sum the two embeddings before the first encoding layer.

All models are trained using Adam [28] as optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. The learning rate is varied during training using the strategy presented in [71], *i.e.* according to the formula: $d^{-0.5} \cdot \min(s^{-0.5}, s \cdot w^{-0.5})$, where s is the current optimization step and w is a warmup parameter, set to 10 000 in all our experiments. After the pre-training with cross-entropy loss, we finetune the models with a fixed learning rate of 5×10^{-6}.

5.4. Captioning performance

We first report the captioning performance of the aforementioned models, when using image region features and when employing the Vision Transformer as visual backbone. Table 1 shows the captioning performance of all models

Table 1
Captioning performance of the considered models, on the COCO Karpathy splits

Method	Validation Set								Test Set							
	B-1	B-4	M	R	C	S	NW	Cov	B-1	B-4	M	R	C	S	NW	Cov
Image Region Features																
Transformer	79.4	36.7	27.7	57.1	122.7	20.7	0.240	74.1	79.4	36.4	27.7	56.8	123.7	20.9	0.242	74.5
SMArT [10]	80.5	**38.3**	**29.0**	**58.4**	**129.3**	22.3	**0.258**	**76.1**	80.4	38.1	28.8	58.2	129.7	22.2	**0.256**	**76.2**
$\mathcal{M}^2$ Transformer [14]	**80.7**	**38.3**	28.9	**58.4**	129.0	**22.4**	0.254	75.9	**80.8**	**39.1**	**29.2**	**58.6**	**131.2**	**22.6**	**0.256**	76.0
Vision Transformer (ViT)																
Transformer	81.0	38.9	28.9	58.6	128.7	22.6	0.261	77.6	80.5	38.5	28.8	58.5	128.7	22.6	0.258	77.4
SMArT [10]	81.1	39.0	29.1	58.6	129.4	22.9	**0.263**	77.1	**81.1**	**38.9**	29.0	58.6	130.7	**23.0**	**0.263**	77.5
$\mathcal{M}^2$ Transformer [14]	**81.2**	**39.2**	**29.2**	**58.7**	**130.3**	**23.0**	0.261	**77.7**	80.8	38.7	**29.2**	**58.7**	**130.9**	**23.0**	**0.263**	**77.9**

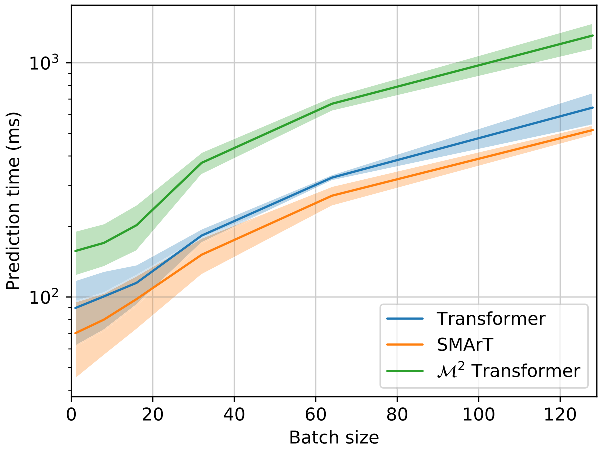


Fig. 2. Prediction times of the considered models when varying the batch size.

on the COCO validation and test splits defined in [27], in terms of standard captioning metrics, NW, and Coverage scores. As it can be noticed, the pre-trained Vision Transformer is competitive with the more standard practice of using image region features in terms of standard captioning metrics, as it advances the performances on the COCO validation set while providing similar performances on the test set. We notice, further, that it brings an improvement in terms of NW and Coverage with respect to the usage of image region features, thus confirming that it can be employed as a proper visual backbone for the generation of quality captions.

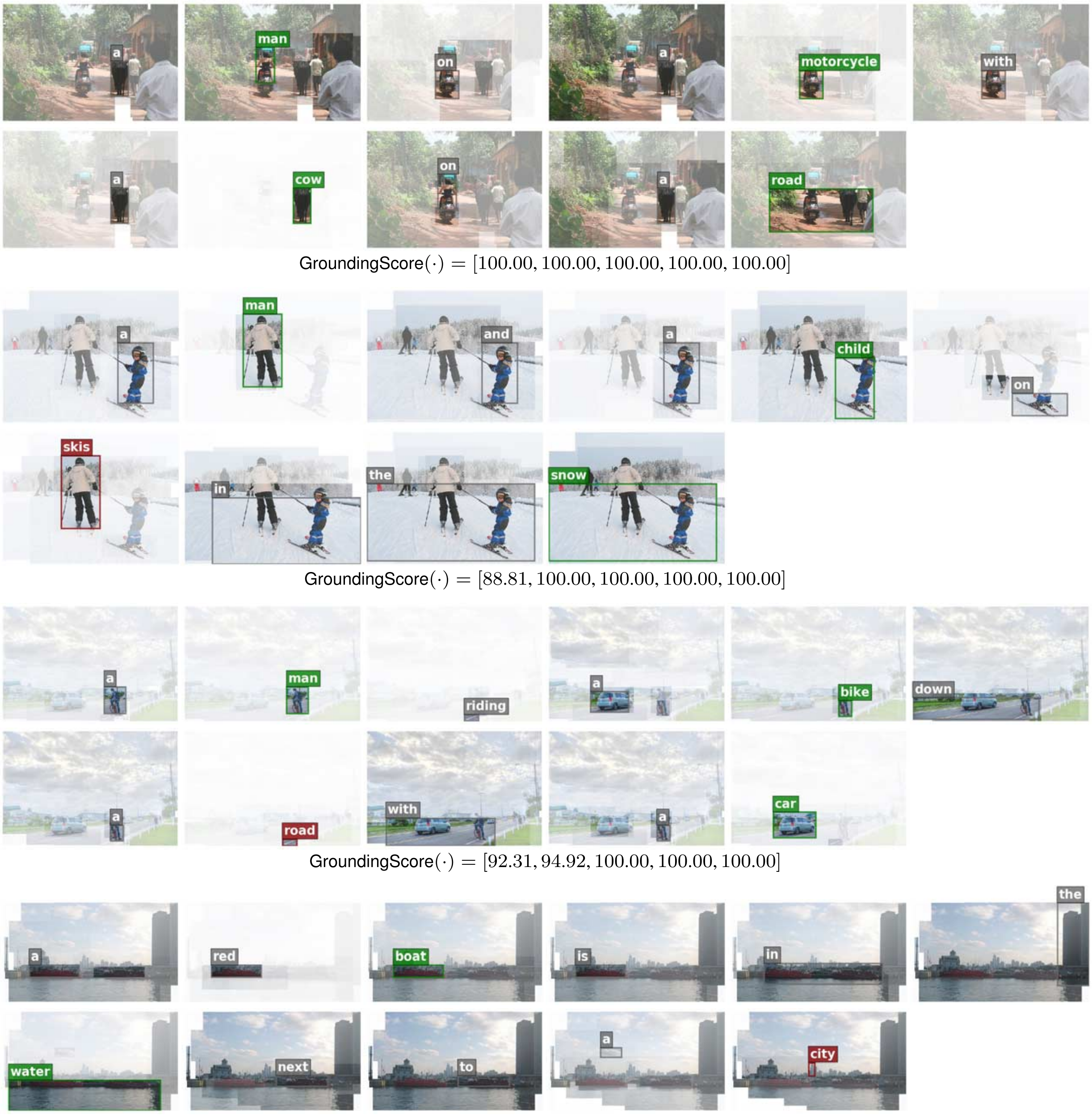
To analyze the computational demands of the three considered captioning approaches, we report in Fig. 2 the mean and standard deviation of prediction times as a function of the batch size. In this analysis, we only evaluate the prediction times of the three captioning architectures without considering the feature extraction phase. For a fair evaluation, we run all approaches on the same workstation and on the same GPU, *i.e.* an NVIDIA 2080Ti. As it can be observed, the SMArT architecture achieves a better efficiency than a traditional Transformer with three layers, thanks to its reduced number of layers (*i.e.* 2). Finally, the $\mathcal{M}^2$ Transformer model has the highest prediction time, because of the meshed connection between encoder and decoder layers.

5.5. *Grounding and alignment evaluation*

Qualitative examples. We now turn to the evaluation of the grounding and alignment capabilities of the models, using the proposed GroundingScore. Figure 3 reports qualitative examples of the metric, computed over sample images and captions generated by the SMArT model. For visualization purposes, the metric is reported as an array, containing the values of GroundingScore at $\delta \in \{0, 1, 3, 5, \infty\}$. As it can be seen, the metric effectively captures the alignment and grounding capabilities of the model. In the first row, the network concentrates on the right object class while generating all nouns of the caption (“man”, “motorcycle”, “cow”, “road”), thus getting a 100% GroundingScore at $\delta = 0$. In the second case, instead, it concentrates on the skis region one step before generating “skis”: because of this slight temporal misalignment, its GroundingScore at $\delta = 0$ is 88.8%, while its GroundingScore at $\delta = 1$ becomes 100%. In the last sample, finally, the GroundingScore never reaches 100%, as the network hallucinates the concept of “city” while just looking at a detection containing a building, and never attends a detection containing the whole skyline. Similarly, in Fig. 4 we report qualitative examples on captions generated by the $\mathcal{M}^2$ Transformer model.

Quantitative evaluation. Tables 2 and 3 reports the GroundingScore obtained for different values of the temporal margin δ on the COCO validation and test sets, respectively. In the same tables, we report the scores obtained by applying all the aforementioned attribution methods, *i.e.* Saliency, Guided Backpropagation, and Integrated Gradients.

Starting from the models employing regions as image descriptors, we notice that both SMArT and $\mathcal{M}^2$ Transformer perform favorably in terms of GroundingScore with respect to the original Transformer, thus confirming that the usage of memory vector and mesh-like connectivity does not only increases quality metrics, but also the



$\mathsf{GroundingScore}(\cdot) = [100.00, 100.00, 100.00, 100.00, 100.00]$

$\mathsf{GroundingScore}(\cdot) = [88.81, 100.00, 100.00, 100.00, 100.00]$

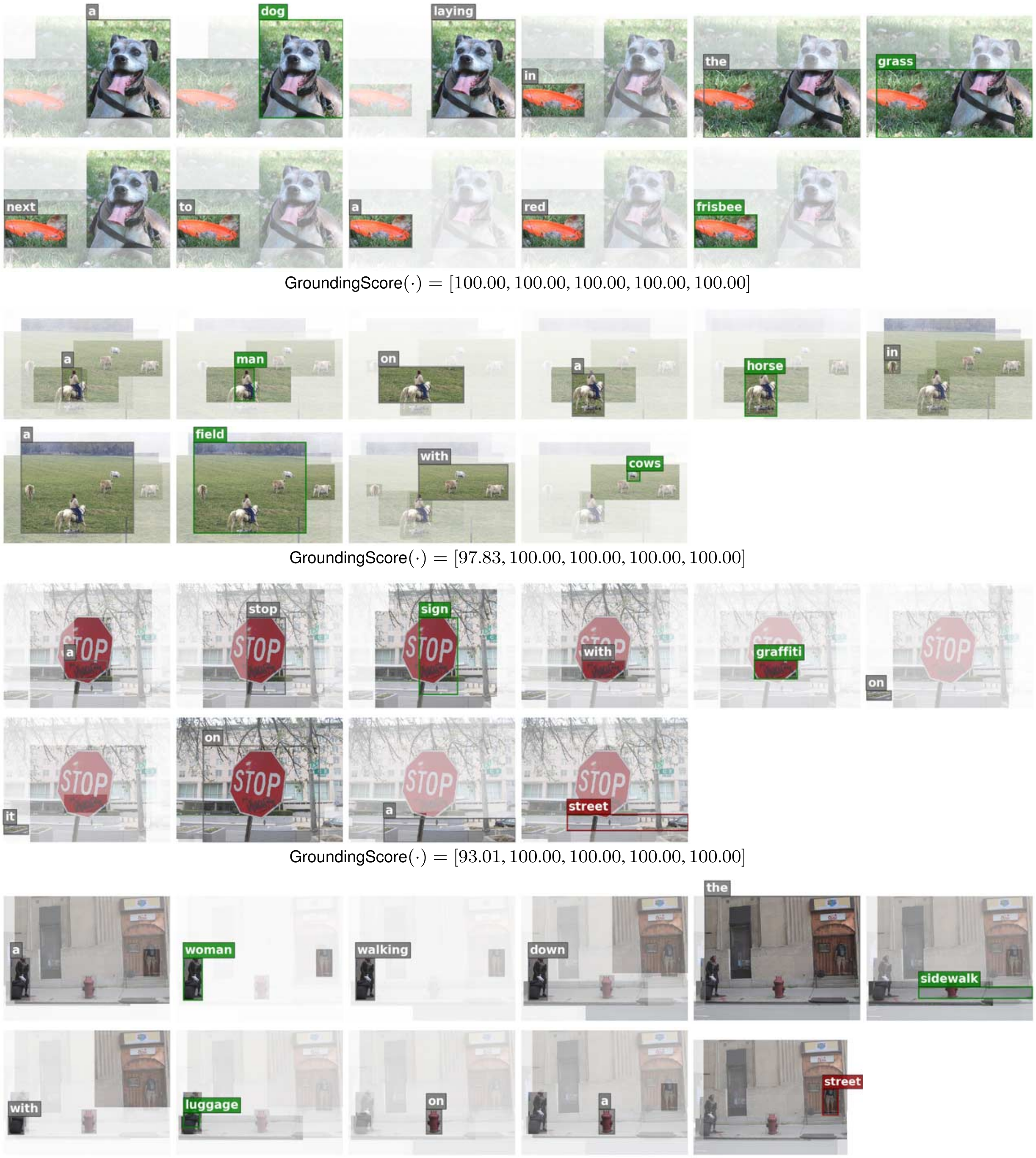
$\mathsf{GroundingScore}(\cdot) = [92.31, 94.92, 100.00, 100.00, 100.00]$

$\mathsf{GroundingScore}(\cdot) = [66.67, 86.97, 88.20, 89.09, 89.09]$

Fig. 3. Qualitative examples of the GroundingScore metric, on captions generated by the SMArT model on COCO images. The metric is reported as an array, containing the values of GroundingScore at $\delta \in \{0, 1, 3, 5, \infty\}$. At each timestep, the detection with the highest attribution score is reported. Correctly grounded nouns, at $\delta = 0$, are reported in green, while incorrectly grounded nouns are reported in red.

grounding and alignment capability of the model. According to all attribution methods, both models start with high scores at $\delta = 0$, thus showing high grounding capabilities and synchronization with respect to attended objects while generating the output caption. For $\delta = \infty$ (*i.e.* when considering an infinite temporal window), they both reach grounding scores between 80 and 88%, depending on the attribution method of choice and the split.

Turning then to the evaluation of the Vision Transformer as visual backbone, we instead notice that it generally lowers the grounding capabilities of the models. The maximum grounding score at $\delta = 0$, for instance, is 70.13,



GroundingScore$(\cdot) = [100.00, 100.00, 100.00, 100.00, 100.00]$

GroundingScore$(\cdot) = [97.83, 100.00, 100.00, 100.00, 100.00]$

GroundingScore$(\cdot) = [93.01, 100.00, 100.00, 100.00, 100.00]$

GroundingScore$(\cdot) = [92.03, 92.03, 92.03, 92.03, 95.76]$

Fig. 4. Qualitative examples of the GroundingScore metric, on captions generated by the $\mathcal{M}^2$ Transformer model on COCO images. The metric is reported as an array, containing the values of GroundingScore at $\delta \in \{0, 1, 3, 5, \infty\}$. At each timestep, the detection with the highest attribution score is reported. Correctly grounded nouns, at $\delta = 0$, are reported in green, while incorrectly grounded nouns are reported in red.

Table 2

Grounding and alignment performance of the considered models on the COCO validation set, expressed in terms of GroundingScore

Method	Saliency					Guided Backprop					Integrated Gradients				
	$\delta = 0$	$\delta = 1$	$\delta = 3$	$\delta = 5$	$\delta = \infty$	$\delta = 0$	$\delta = 1$	$\delta = 3$	$\delta = 5$	$\delta = \infty$	$\delta = 0$	$\delta = 1$	$\delta = 3$	$\delta = 5$	$\delta = \infty$
Image Region Features															
Transformer	75.15	78.59	80.28	80.84	81.20	67.42	74.23	78.37	80.00	80.78	79.26	82.76	84.81	85.43	85.78
SMArT [10]	80.37	83.76	85.68	86.15	86.36	**72.23**	**78.39**	**81.75**	**82.87**	**83.51**	**81.30**	**85.36**	**87.47**	**87.92**	**88.12**
$\mathcal{M}^2$ Transformer [14]	**80.68**	**84.01**	**85.85**	**86.26**	**86.45**	71.43	77.98	81.53	82.68	83.28	75.16	80.90	83.83	84.71	85.20
Vision Transformer (ViT)															
Transformer	69.94	70.79	71.44	71.67	71.83	67.19	73.59	77.12	78.48	79.30	69.02	75.38	78.90	80.14	80.95
SMArT [10]	70.08	70.83	71.40	71.62	71.75	67.39	74.21	77.62	78.93	79.78	69.42	75.85	79.27	80.52	81.25
$\mathcal{M}^2$ Transformer [14]	**70.13**	**71.02**	**71.62**	**71.84**	**71.97**	**67.85**	**74.31**	**77.87**	**79.10**	**79.93**	**69.56**	**75.90**	**79.48**	**80.70**	**81.46**

Table 3

Grounding and alignment performance of the considered models on the COCO test set, expressed in terms of GroundingScore

Method	Saliency					Guided Backprop					Integrated Gradients				
	$\delta = 0$	$\delta = 1$	$\delta = 3$	$\delta = 5$	$\delta = \infty$	$\delta = 0$	$\delta = 1$	$\delta = 3$	$\delta = 5$	$\delta = \infty$	$\delta = 0$	$\delta = 1$	$\delta = 3$	$\delta = 5$	$\delta = \infty$
Image Region Features															
Transformer	74.95	78.18	80.06	80.56	80.84	67.02	74.08	78.25	79.88	80.61	79.22	82.75	84.86	85.43	85.77
SMArT [10]	79.83	83.45	85.57	85.98	86.17	**72.28**	**78.15**	**81.60**	**82.79**	**83.37**	**80.66**	**84.98**	**87.20**	**87.69**	**87.89**
$\mathcal{M}^2$ Transformer [14]	**80.23**	**83.70**	**85.74**	**86.12**	**86.28**	71.29	78.07	**81.60**	82.67	83.24	74.98	80.60	83.69	84.54	84.98
Vision Transformer (ViT)															
Transformer	69.89	70.69	71.34	71.56	71.68	67.14	73.54	77.04	78.43	79.18	69.11	75.55	78.93	80.20	80.94
SMArT [10]	69.96	70.81	**71.41**	71.59	71.72	67.31	**74.12**	77.65	79.01	79.83	69.85	75.73	79.24	80.43	81.09
$\mathcal{M}^2$ Transformer [14]	**69.99**	**70.85**	71.37	**71.62**	**71.76**	**67.54**	**74.12**	**77.81**	**79.24**	**80.06**	**69.86**	**76.01**	**79.55**	**80.72**	**81.31**

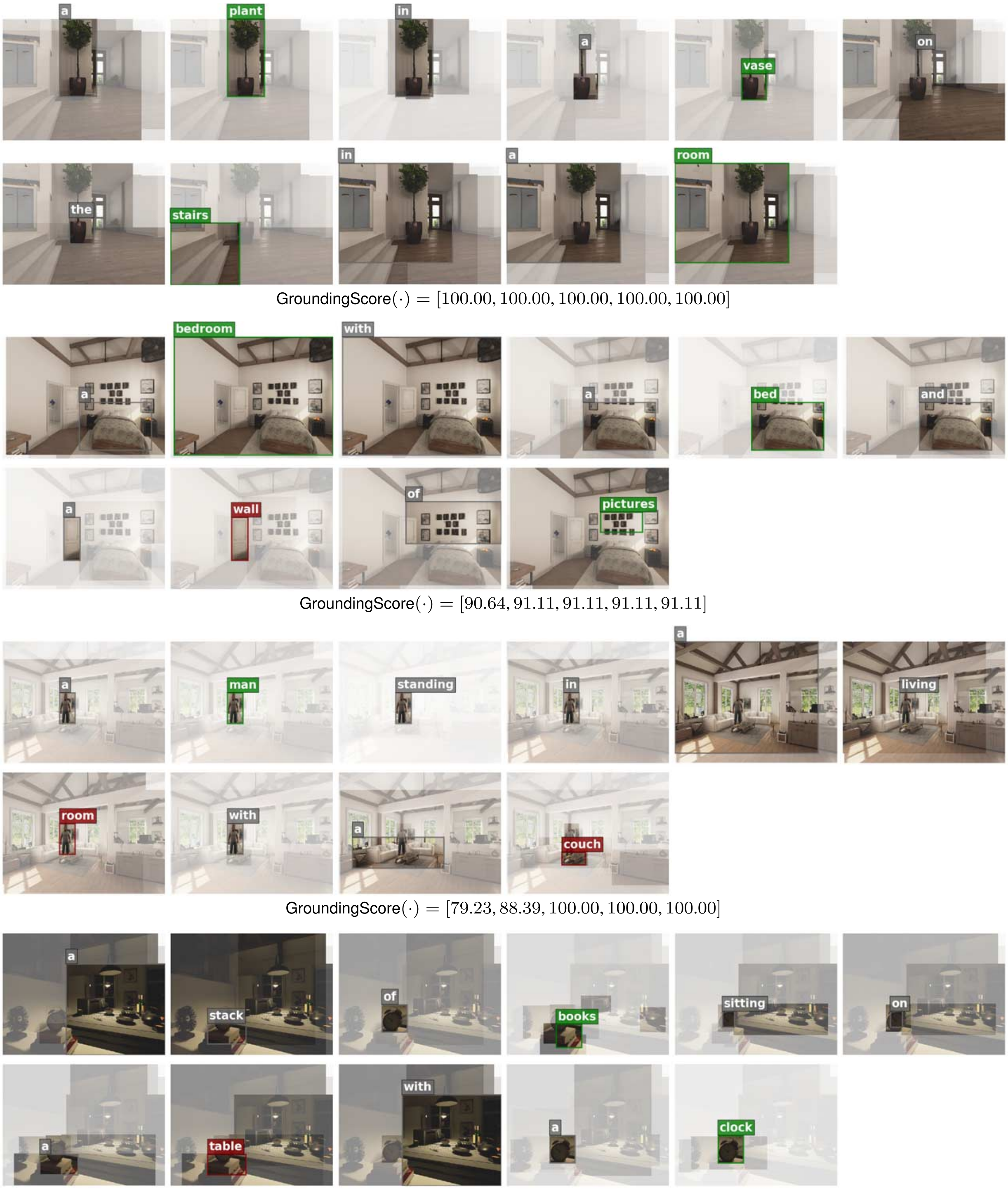
Table 4

Grounding and alignment performance of the considered models on the ACRV dataset, expressed in terms of GroundingScore

Method	Saliency					Guided Backprop					Integrated Gradients				
	$\delta = 0$	$\delta = 1$	$\delta = 3$	$\delta = 5$	$\delta = \infty$	$\delta = 0$	$\delta = 1$	$\delta = 3$	$\delta = 5$	$\delta = \infty$	$\delta = 0$	$\delta = 1$	$\delta = 3$	$\delta = 5$	$\delta = \infty$
Image Region Features															
Transformer	74.91	78.39	79.99	80.40	80.62	68.02	76.12	79.83	81.02	81.47	76.91	81.84	83.92	84.44	84.66
SMArT [10]	80.56	84.29	85.92	86.41	86.57	**74.80**	**80.67**	**84.02**	**84.95**	**85.37**	79.42	**86.41**	**88.13**	**88.83**	**89.08**
$\mathcal{M}^2$ Transformer [14]	**82.13**	**85.77**	**87.33**	**87.81**	**88.00**	72.36	78.84	82.46	83.44	83.86	**79.52**	84.53	86.38	87.03	87.40
Vision Transformer (ViT)															
Transformer	71.03	72.90	75.28	77.20	**80.00**	72.72	75.83	77.76	**78.88**	79.92	71.66	**74.74**	**77.46**	**78.84**	**79.95**
SMArT [10]	**71.14**	**72.91**	75.36	77.21	79.91	72.26	75.47	77.51	78.65	79.65	71.51	74.41	76.91	71.92	79.54
$\mathcal{M}^2$ Transformer [14]	71.13	72.86	**75.77**	**77.29**	**80.00**	**72.83**	**75.94**	**77.96**	**78.88**	**80.01**	**71.73**	74.69	77.15	78.21	79.40

thus significantly lower than the average score obtained when employing region features. Also, we notice that the relative difference between the grounding score at $\delta = \infty$ and $\delta = 0$ reduces, indicating that the Vision Transformer fosters hallucination and incorrect localization rather than generating synchronization flaws.

Finally, in Table 4 we report the same analysis on the ACVR dataset, which features simulated data and a robot-centric point of view. As it can be seen, there is no reduction in grounding capabilities, as testified by the values reported according to all attribution methods. Instead, we observe a slight increase of the grounding metric at $\delta = 0$, which underlines that the models can ground objects correctly also in this simulated setting.



GroundingScore$(\cdot) = [100.00, 100.00, 100.00, 100.00, 100.00]$

GroundingScore$(\cdot) = [90.64, 91.11, 91.11, 91.11, 91.11]$

GroundingScore$(\cdot) = [79.23, 88.39, 100.00, 100.00, 100.00]$

GroundingScore$(\cdot) = [84.99, 84.99, 87.12, 87.12, 89.28]$

Fig. 5. Qualitative examples of the GroundingScore metric, on captions generated on the ACVR dataset. The metric is reported as an array, containing the values of GroundingScore at $\delta \in \{0, 1, 3, 5, \infty\}$. At each timestep, the detection with the highest attribution score is reported. Correctly grounded nouns, at $\delta = 0$, are reported in green, while incorrectly grounded nouns are reported in red.

6. Conclusion

We have presented a visualization and grounding methodology for Transformer-based image captioning algorithms. Our work takes inspiration from the increasing need of explaining models' predictions and quantify their grounding capabilities. Given the highly non-linear nature of state-of-the-art captioners, we employed different attribution methods to visualize what the model concentrates on at each step of the generation. Further, we have proposed a metric that quantifies the grounding and temporal alignment capabilities of a model, and which can be used to measure hallucination or synchronization flaws. Experiments have been conducted on the COCO and ACVR datasets, employing three Transformer-based architectures.

Acknowledgements

This work has been supported by "Fondazione di Modena", by the "Artificial Intelligence for Cultural Heritage (AI4CH)" project, co-funded by the Italian Ministry of Foreign Affairs and International Cooperation, and by the H2020 ICT-48-2020 HumanE-AI-NET and H2020 ECSEL "InSecTT – Intelligent Secure Trustable Things" projects. InSecTT (www.insectt.eu) has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876038. The JU receives support from the European Union's Horizon 2020 research and innovation program and Austria, Sweden, Spain, Italy, France, Portugal, Ireland, Finland, Slovenia, Poland, Netherlands, Turkey. The document reflects only the authors' view and the Commission is not responsible for any use that may be made of the information it contains.

References

[1] P. Anderson, B. Fernando, M. Johnson and S. Gould, SPICE: Semantic propositional image caption evaluation, in: *Proceedings of the European Conference on Computer Vision*, 2016.

[2] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould and L. Zhang, Bottom-up and top-down attention for image captioning and visual question answering, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[3] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould and A. van den Hengel, Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[4] J. Aneja, A. Deshpande and A.G. Schwing, Convolutional image captioning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[5] J.L. Ba, J.R. Kiros and G.E. Hinton, Layer normalization, 2016, arXiv preprint arXiv:1607.06450.

[6] D. Bahdanau, K. Cho and Y. Bengio, Neural machine translation by jointly learning to align and translate, in: *Proceedings of the International Conference on Learning Representations*, 2014.

[7] S. Banerjee and A. Lavie, METEOR: An automatic metric for MT evaluation with improved correlation with human judgments, in: *Proceedings of the Annual Meeting of the Association for Computational Linguistics Workshops*, 2005.

[8] M. Cagrandi, M. Cornia, M. Stefanini, L. Baraldi and R. Cucchiara, Learning to select: A fully attentive approach for novel object captioning, in: *ICMR*, 2021.

[9] M. Cornia, L. Baraldi and R. Cucchiara, Show, control and tell: A framework for generating controllable and grounded captions, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[10] M. Cornia, L. Baraldi and R. Cucchiara, SMArT: Training shallow memory-aware transformers for robotic explainability, in: *Proceedings of the IEEE International Conference on Robotics and Automation*, 2020.

[11] M. Cornia, L. Baraldi, G. Serra and R. Cucchiara, SAM: Pushing the limits of saliency prediction models, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[12] M. Cornia, L. Baraldi, G. Serra and R. Cucchiara, Paying more attention to saliency: Image captioning with saliency and context attention, *ACM Transactions on Multimedia Computing, Communications, and Applications* **14**(2) (2018), 1–21. doi:10.1145/3177745.

[13] M. Cornia, L. Baraldi, H.R. Tavakoli and R. Cucchiara, A unified cycle-consistent neural model for text and image retrieval, *Multimedia Tools and Applications* **79**(35) (2020), 25697–25721. doi:10.1007/s11042-020-09251-4.

[14] M. Cornia, M. Stefanini, L. Baraldi and R. Cucchiara, Meshed-memory transformer for image captioning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[15] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2018.

[16] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko and T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015.

[17] 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, in: *Proceedings of the International Conference on Learning Representations*, 2021.

[18] F. Faghri, D.J. Fleet, J.R. Kiros and S. Fidler, VSE++: Improving visual-semantic embeddings with hard negatives, in: *Proceedings of the British Machine Vision Conference*, 2018.

[19] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Prentice-Hall, 2002.

[20] L. Guo, J. Liu, X. Zhu, P. Yao, S. Lu and H. Lu, Normalized and geometry-aware self-attention network for image captioning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[21] D. Hall, F. Dayoub, J. Skinner, H. Zhang, D. Miller, P. Corke, G. Carneiro, A. Angelova and N. Sünderhauf, Probabilistic object detection: Definition and evaluation, in: *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2020.

[22] K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[23] S. Herdade, A. Kappeler, K. Boakye and J. Soares, Image captioning: Transforming objects into words, in: *Advances in Neural Information Processing Systems*, 2019.

[24] L. Huang, W. Wang, J. Chen and X.-Y. Wei, Attention on attention for image captioning, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[25] J. Ji, Y. Luo, X. Sun, F. Chen, G. Luo, Y. Wu, Y. Gao and R. Ji, Improving image captioning by leveraging intra- and inter-layer global representation in transformer network, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[26] J. Johnson, A. Karpathy and L. Fei-Fei, DenseCap: Fully convolutional localization networks for dense captioning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[27] A. Karpathy and L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015.

[28] D.P. Kingma and J. Ba, Adam: A method for stochastic optimization, in: *Proceedings of the International Conference on Learning Representations*, 2015.

[29] J. Krantz, E. Wijmans, A. Majumdar, D. Batra and S. Lee, Beyond the nav-graph: Vision-and-language navigation in continuous environments, in: *Proceedings of the European Conference on Computer Vision*, 2020.

[30] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D.A. Shamma, M. Bernstein and L. Fei-Fei, Visual genome: Connecting language and vision using crowdsourced dense image annotations, *International Journal of Computer Vision* **123**(1) (2017), 32–73. doi:10.1007/s11263-016-0981-7.

[31] H.W. Kuhn, The Hungarian method for the assignment problem, *Naval Research Logistics Quarterly* **2**(1–2) (1955), 83–97. doi:10.1002/nav.3800020109.

[32] F. Landi, L. Baraldi, M. Cornia, M. Corsini and R. Cucchiara, Multimodal attention networks for low-level vision-and-language navigation, *Computer Vision and Image Understanding* (2021).

[33] F. Landi, L. Baraldi, M. Cornia and R. Cucchiara, Working memory connections for LSTM, *Neural Networks* **144** (2021), 334–341. doi:10.1016/j.neunet.2021.08.030.

[34] K.-H. Lee, X. Chen, G. Hua, H. Hu and X. He, Stacked cross attention for image-text matching, in: *Proceedings of the European Conference on Computer Vision*, 2018.

[35] G. Li, L. Zhu, P. Liu and Y. Yang, Entangled transformer for image captioning, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[36] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei et al., Oscar: Object-semantics aligned pre-training for vision-language tasks, in: *Proceedings of the European Conference on Computer Vision*, 2020.

[37] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: *Proceedings of the Annual Meeting of the Association for Computational Linguistics Workshops*, 2004.

[38] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C.L. Zitnick, Microsoft COCO: Common objects in context, in: *Proceedings of the European Conference on Computer Vision*, 2014.

[39] S. Liu, Z. Zhu, N. Ye, S. Guadarrama and K. Murphy, Improved image captioning via policy gradient optimization of SPIDEr, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

[40] W. Liu, S. Chen, L. Guo, X. Zhu and J. Liu, CPTR: Full transformer network for image captioning, 2021, arXiv preprint arXiv:2101.10804.

[41] J. Lu, D. Batra, D. Parikh and S. Lee, ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, in: *Advances in Neural Information Processing Systems*, 2019.

[42] J. Lu, C. Xiong, D. Parikh and R. Socher, Knowing when to look: Adaptive attention via a visual sentinel for image captioning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[43] J. Lu, J. Yang, D. Batra and D. Parikh, Neural baby talk, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[44] C.-Y. Ma, Y. Kalantidis, G. AlRegib, P. Vajda, M. Rohrbach and Z. Kira, Learning to generate grounded visual captions without localization supervision, in: *Proceedings of the European Conference on Computer Vision*, 2020.

[45] S.B. Needleman and C.D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, *Journal of Molecular Biology* **48**(3) (1970), 443–453. doi:10.1016/0022-2836(70)90057-4.

[46] Y. Pan, T. Yao, Y. Li and T. Mei, X-linear attention networks for image captioning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[47] K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, BLEU: A method for automatic evaluation of machine translation, in: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2002.

[48] M. Pedersoli, T. Lucas, C. Schmid and J. Verbeek, Areas of attention for image captioning, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

[49] J. Pennington, R. Socher and C.D. Manning, GloVe: Global vectors for word representation, in: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2014.

[50] S. Poppi, M. Cornia, L. Baraldi and R. Cucchiara, Revisiting the evaluation of class activation mapping for explainability: A novel metric and experimental analysis, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2021.

[51] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger and I. Sutskever, Learning transferable visual models from natural language supervision, 2021, arXiv preprint arXiv:2103.00020.

[52] V. Ramanishka, A. Das, J. Zhang and K. Saenko, Top-down visual saliency guided by captions, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[53] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen and I. Sutskever, Zero-shot text-to-image generation, 2021, arXiv preprint arXiv:2102.12092.

[54] M. Ranzato, S. Chopra, M. Auli and W. Zaremba, Sequence level training with recurrent neural networks, in: *Proceedings of the International Conference on Learning Representations*, 2016.

[55] S. Ren, K. He, R. Girshick and J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: *Advances in Neural Information Processing Systems*, 2015.

[56] S.J. Rennie, E. Marcheret, Y. Mroueh, J. Ross and V. Goel, Self-critical sequence training for image captioning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[57] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

[58] S. Shen, L.H. Li, H. Tan, M. Bansal, A. Rohrbach, K.-W. Chang, Z. Yao and K. Keutzer, How much can clip benefit vision-and-language tasks? 2021, arXiv preprint arXiv:2107.06383.

[59] K. Simonyan, A. Vedaldi and A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, 2013, arXiv preprint arXiv:1312.6034.

[60] R. Socher and L. Fei-Fei, Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2010.

[61] J.T. Springenberg, A. Dosovitskiy, T. Brox and M. Riedmiller, Striving for simplicity: The all convolutional net, 2014, arXiv preprint arXiv:1412.6806.

[62] M. Stefanini, M. Cornia, L. Baraldi, S. Cascianelli, G. Fiameni and R. Cucchiara, From show to tell: A survey on image captioning, 2021, arXiv preprint arXiv:2107.06912.

[63] M. Stefanini, M. Cornia, L. Baraldi and R. Cucchiara, A novel attention-based aggregation function to combine vision and language, in: *Proceedings of the International Conference on Pattern Recognition*, 2020.

[64] Y. Sugano and A. Bulling, Seeing with humans: Gaze-assisted neural image captioning, 2016, arXiv preprint arXiv:1608.05203.

[65] S. Sukhbaatar, E. Grave, G. Lample, H. Jegou and A. Joulin, Augmenting self-attention with persistent memory, 2019, arXiv preprint arXiv:1907.01470.

[66] J. Sun, S. Lapuschkin, W. Samek and A. Binder, Explain and improve: LRP-inference fine-tuning for image captioning models, *Information Fusion* (2021).

[67] M. Sundararajan, A. Taly and Q. Yan, Axiomatic attribution for deep networks, in: *Proceedings of the International Conference on Machine Learning*, 2017.

[68] H. Tan and M. Bansal, Lxmert: Learning cross-modality encoder representations from transformers, in: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2019.

[69] H.R. Tavakoli, R. Shetty, A. Borji and J. Laaksonen, Paying attention to descriptions generated by image captioning models, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

[70] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles and H. Jégou, Training data-efficient image transformers & distillation through attention, in: *Proceedings of the International Conference on Machine Learning*, 2021.

[71] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser and I. Polosukhin, Attention is all you need, in: *Advances in Neural Information Processing Systems*, 2017.

[72] R. Vedantam, C. Lawrence Zitnick and D. Parikh, CIDEr: Consensus-based image description evaluation, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015.

[73] O. Vinyals, A. Toshev, S. Bengio and D. Erhan, Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**(4) (2016), 652–663. doi:10.1109/TPAMI.2016.2587640.

[74] W. Wang and J. Shen, Deep visual attention prediction, *IEEE Transactions on Image Processing* **27**(5) (2017), 2368–2378. doi:10.1109/TIP.2017.2787612.

[75] Z. Wang, J. Yu, A.W. Yu, Z. Dai, Y. Tsvetkov and Y. Cao, SimVLM: Simple visual language model pretraining with weak supervision, 2021, arXiv preprint arXiv:2108.10904.

[76] Q. Xia, H. Huang, N. Duan, D. Zhang, L. Ji, Z. Sui, E. Cui, T. Bharti and M. Zhou, XGPT: Cross-modal generative pre-training for image captioning, 2020, arXiv preprint arXiv:2003.01473.

[77] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R.S. Zemel and Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: *Proceedings of the International Conference on Machine Learning*, 2015.

[78] X. Yang, K. Tang, H. Zhang and J. Cai, Auto-encoding scene graphs for image captioning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[79] X. Yang, H. Zhang and J. Cai, Learning to collocate neural modules for image captioning, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[80] B.Z. Yao, X. Yang, L. Lin, M.W. Lee and S.-C. Zhu, I2t: Image parsing to text description, *Proceedings of the IEEE* (2010).

[81] T. Yao, Y. Pan, Y. Li and T. Mei, Exploring visual relationship for image captioning, in: *Proceedings of the European Conference on Computer Vision*, 2018.

[82] Q. You, H. Jin, Z. Wang, C. Fang and J. Luo, Image captioning with semantic attention, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[83] P. Zhang, X. Li, X. Hu, J. Yang, L. Zhang, L. Wang, Y. Choi and J. Gao, VinVL: Revisiting visual representations in vision-language models, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[84] L. Zhou, Y. Kalantidis, X. Chen, J.J. Corso and M. Rohrbach, Grounded video description, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[85] L. Zhou, H. Palangi, L. Zhang, H. Hu, J.J. Corso and J. Gao, Unified vision-language pre-training for image captioning and VQA, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.