Connecting What to Say With Where to Look by Modeling Human Attention Traces

Zihang Meng[1], Licheng Yu[2], Ning Zhang[2], Tamara Berg[2],
Babak Damavandi[2], Vikas Singh[1], and Amy Bearman[2]

[1]University of Wisconsin Madison [2]Facebook AI
zmeng29@wisc.edu, vsingh@biostat.wisc.edu
{lichengyu,ningzhang,tlberg,babakd,abearman}@fb.com

Abstract

We introduce a unified framework to jointly model images, text, and human attention traces. Our work is built on top of the recent Localized Narratives annotation framework [31], where each word of a given caption is paired with a mouse trace segment. We propose two novel tasks: (1) predict a trace given an image and caption (i.e., visual grounding), and (2) predict a caption and a trace given only an image. Learning the grounding of each word is challenging, due to noise in the human-provided traces and the presence of words that cannot be meaningfully visually grounded. We present a novel model architecture that is jointly trained on dual tasks (controlled trace generation and controlled caption generation). To evaluate the quality of the generated traces, we propose a local bipartite matching (LBM) distance metric which allows the comparison of two traces of different lengths. Extensive experiments show our model is robust to the imperfect training data and outperforms the baselines by a clear margin. Moreover, we demonstrate that our model pre-trained on the proposed tasks can be also beneficial to the downstream task of COCO's guided image captioning. Our code[1] and project page[2] are publicly available.

1. Introduction

The development of powerful models and algorithms within computer vision and natural language processing proceeded along distinct trajectories with only occasional overlap until recently. However, ideas from these two fields are gradually converging, with a focus on building multimodal models, particularly for aligning visual and language stimuli [25, 35, 34, 10]. The goal of these models is to mimic humans' extraordinary abilities to compress information and translate it across modalities. Several joint or combined visual recognition and natural language understanding tasks have emerged as natural tests of these vision-and-language models' capabilities. *Image captioning* asks a model to identify and localize the key scene elements in an image and describe them in natural language form. *Visual grounding*, and specifically *phrase localization*, asks a model to solve the reverse problem: given a natural language query, identify the target object(s) of the query in the image. *Controlled image captioning*, first introduced in [12], combines the two tasks. Here, an external user is asked to specify which parts of the image they want described and in what order (e.g., by providing an ordered sequence of bounding boxes). The output captions are therefore explicitly grounded in the image. One application of this line of work is automatically generating localized descriptions of images for visually impaired users on social

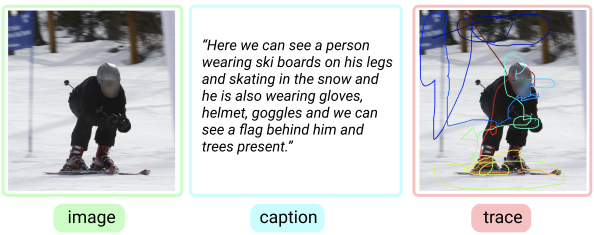

Figure 1: The three vision-and-language tasks, as illustrated on a single example from the Localized Narratives dataset. The first and third depicted tasks are novel.

[1]Code: github.com/facebookresearch/connect-caption-and-trace
[2]Project page: http://pages.cs.wisc.edu/~zihangm/connect_caption_trace

media services. This removes the need to rely on human-written "alt" text, which is often missing in web images [6].

Vision-and-language models share common components and techniques. Image captioning architectures are typically composed of two modules: an image encoder, which ingests and interprets an image, and a language model decoder, which generates a natural language caption [39, 17]. Visual grounding models first identify the key components of the image (i.e., bounding box proposals) and query (i.e., which words or phrases to focus on), extract features from each, and then correlate them to predict the referred-to object [33, 16, 30, 42]. Architectures for both tasks often rely on attention [39, 15, 17, 25], a mechanism inspired by the human visual system [32, 11]. Researchers have also designed more complex models that can do both caption generation and grounding. For example, [27] and [42] can both generate an unambiguous description of a specific object or region in an image and automatically select an object given a referring text expression.

Despite these advancements, existing image captioning and visual grounding models cannot jointly generate long-form, natural language captions *and* dense, word-level visual groundings. This is because existing image captioning datasets only provide short captions with sparse groundings at the noun level (Flickr30k Entities [30]) or phrase level (Google RefEx [27], Flickr30k Entities [30] and Visual Genome [22]). To address these limitations, [31] introduced the Localized Narratives dataset, in which annotators were asked to describe an image with their voice while simultaneously drawing a mouse trace over the region they are describing. This annotation framework provides rich, longform image captions and dense visual grounding in the form of a mouse trace segment for each word. The work in [31] incorporates the annotated mouse trace to aid in standard image captioning and controlled image captioning tasks. However, it does not investigate the reverse problem of directly predicting the mouse trace or explore the connections between caption generation and trace generation.

In this paper, we take a step beyond [31] by requiring models to directly predict the trace, which is analogous to a fine-grained and temporally grounded log of human attention. Besides *controlled caption generation*, where a model generates a caption guided by the given ordered trace from [31], we further introduce two challenging new tasks: *controlled trace generation*, where a model must densely localize each word from a natural language caption in an image, and *joint caption and trace generation*, where a model is only given an image and must act as an annotator in the Localized Narratives protocol. There tasks are shown in Fig. 1. The task of predicting the trace is meaningful in two ways. First, a point-wise trace is a straightforward means for representing eye gaze. Learning the trace (independent of specific use cases) could be variously useful, and this is made possible by the efficient collection scheme described in [31] which does not rely on expensive gaze trackers. Second, this form of annotation yields "weakly-labeled" word-level grounding. We demonstrate that such "weak" word-to-trace alignment could offer benefits for some important vision and language tasks. Besides, the predicted trace can provide a better explanation than most attention-based image captioning approaches. To evaluate the generated traces, we propose a novel evaluation metric, local bipartite matching (LBM), to compare two traces of arbitrary length. We present a flexible new transformer-based model architecture that is trained in parallel on controlled caption generation and controlled trace generation. The model also incorporates a symmetric cycle loss to improve the quality of the generated caption and trace. In addition to the three tasks mentioned above, we show that our approach can benefit downstream tasks by pre-training on our proposed tasks before fine-tuning for the downstream setting.

To summarize, we make the following contributions:

- We introduce two novel tasks: (i) controlled trace generation and (ii) joint caption and trace generation.
- We present a novel mirrored transformer model architecture (MITR), which is jointly trained and evaluated on three vision-and-language tasks.
- We design an evaluation metric to address the challenge of computing the distance between two traces of different lengths.
- By jointly learning from the mirrored trace generation task, our proposed method benefits the downstream task of guided caption generation on the COCO dataset.

2. Related Work

Image Captioning Image captioning is typically formulated using a generative model, creating descriptions in textual space given the input image via CNN-to-RNN/LSTM/Transformer [38, 9, 20, 13]. An increasingly common addition to this basic architecture is a visual attention mechanism which typically produces a spatial map that identifies the specific image region(s) most relevant to the current word prediction task [40, 3]. However, the learned spatial attention may not well align with human attention [14]. To model attention more directly, controlled image captioning was first introduced in [39]. It requires the user to provide a sequence of bounding boxes in the image and outputs the image caption in the same order, describing the objects in those bounding boxes. The authors in [31] adjusted the task by using an annotator's mouse trace for the control.

Visual Grounding The task of visual grounding is to localize a region described by a given text query. Re-

searchers have introduced multiple datasets to tackle this problem, such as RefCOCO [42], Google RefEx [27], Flickr30K [30], and DenseCap [19]. State-of-the-art approaches [41, 43, 24, 44] treat visual grounding as selecting the most matched box to the input text query. However, the input query is typically short (the average length of captions in RefCOCO is 3.5 words) and the grounding is sparse (each query corresponds to just a single box). By contrast, our work focuses on denser word-to-region grounding.

Localized Narratives As described above, image captioning datasets only provide image-sentence pairs without the spatial localization of words. Visual grounding datasets only provide sparse region-sentence mapping. Recently, Localized Narratives [31] was proposed, which offers dense word-region alignment for each full caption. This dataset was collected by recording annotators' voice and mouse traces simultaneously when describing the image content. The three modalities of image, trace, and caption significantly expands the scope of how we can connect vision and language. While [31] only addressed a single task of controlled captioning, we introduce two more novel and challenging tasks, i.e., controlled trace generation, and joint caption and trace generation. At first glance, these three tasks as shown in Fig. 1 appear separate; however, we propose a unified framework using a mirrored transformer to jointly model all three tasks.

3. Method

3.1. Three Tasks on the Three Modalities

We first introduce how we encode the new trace modality. The trace is a series of points with corresponding timestamps, each associated with a single word from the caption. Instead of encoding each individual point, we convert the trace into a sequence of word-aligned bounding boxes, i.e., one box per word. This encoding mitigates the local "drawing" variation (by different annotators) within the same region, and thus more reliably allows the model to attend to the full spatial extent of a referred region.

In order to generate this dense word-to-box alignment from the provided trace points, we take the following steps: (i) Split the trace into segments, with one segment per word (using the word-to-trace alignment from Localized Narratives). (ii) Generate one bounding box per trace segment, by taking the axis-aligned minimum bounding box of the convex hull of the mouse points. Then we introduce the three tasks:

Controlled Trace Generation Given an image I and a caption describing this image $\mathbf{w} = \{w_1, w_2, \cdots, w_N\}$, the model is required to generate a trace indicating the visual grounding corresponding to the caption, in the form of an ordered region sequence $\mathbf{r} = \{r_1, r_2, \cdots, r_T\}$.

Controlled Caption Generation Given an image I and a mouse trace provided by the user that is mapped to a sequence of regions, $\mathbf{r} = \{r_1, r_2, \cdots, r_T\}$, the model generates a caption $\mathbf{w} = \{w_1, w_2, \cdots, w_N\}$ describing the image along this trace.

Joint Caption and Visual Trace Generation We further propose a task which can be regarded as an extension of standard image captioning: given an image I, the model generates both caption $\mathbf{w} = \{w_1, w_2, ..., w_N\}$ and its corresponding trace of ordered regions $\mathbf{r} = \{r_1, r_2..., r_T\}$ that matches the caption.

3.2. Mirrored Transformer for Three Modalities

Although the three tasks defined above are quite different, they operate on the same set of three modalities: image, caption, and trace. In this work, we propose a model that effectively addresses all three tasks together in a unified framework with shared parameters, rather than building three separate models. Due to its symmetric structure, we name this model architecture "**MI**rrored **T**ransforme**R**" (MITR), as in Fig. 2.

Features The inputs to the model are subsets of: image features, text features, and trace features. For image features, we use pre-trained Faster R-CNN [3] to compute the visual features of the detected regions. For the text feature, we sum up the positional embeddings and the word embeddings, as in [36], where the position refers to the index of the word within the caption. For the trace feature, we sum up the positional embeddings and the input trace, which is projected into d hidden dimensions. Specifically, we define the trace position as the index of the bounding box that is aligned with the word in the corresponding caption. We denote the input visual features, text features, and trace features as x_v, x_w, x_r, respectively.

Model Architecture As in Fig. 3, our model is composed of three modules (corresponding to three modalities): image encoder, caption encoder-decoder, and trace encoder-decoder. Each module consists of a transformer with self-attention. Specifically, the image encoder, h_v, is defined as:

$$h_v = \text{FFN}(\text{MultiHead}_v(x_v, x_v, x_v)), \tag{1}$$

where we follow [36] to define the feed-forward network (FFN) as two linear transformation layers with a ReLU activation in between, and the MultiHead as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_c)W^O$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V),$$

where the projections are parameter matrices. We refer readers to [36] for additional details of MultiHead attention. Note that there is no masking operation in the MultiHead module from Eqn. (1), since we allow the model to attend to all visual features when processing the caption and trace.

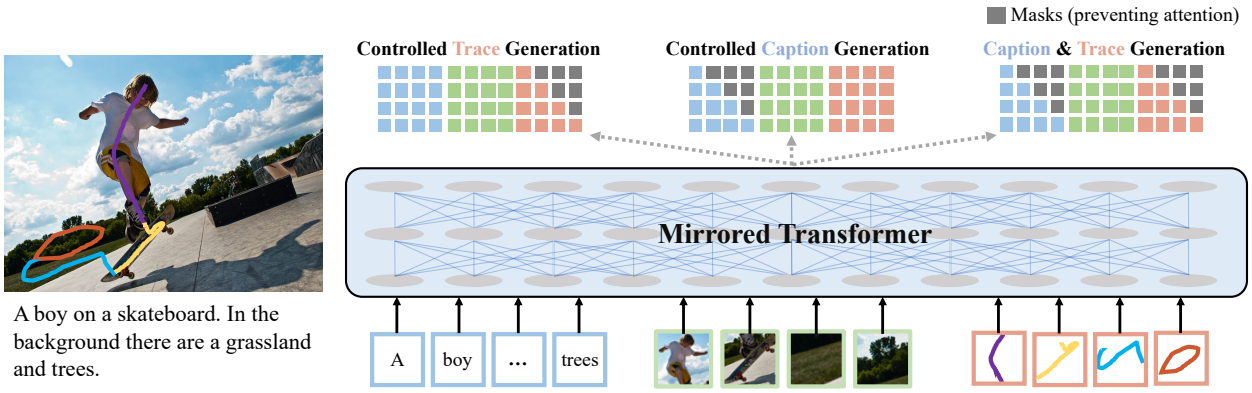

Figure 2: Overall architecture. Our proposed Mirrored Transformer (MITR) architecture effectively addresses the three tasks together by sharing most of the network modules. The structure is mirrored for processing the caption and trace. Depending on the task, we add a masking operation for the encoding/decoding of each module.

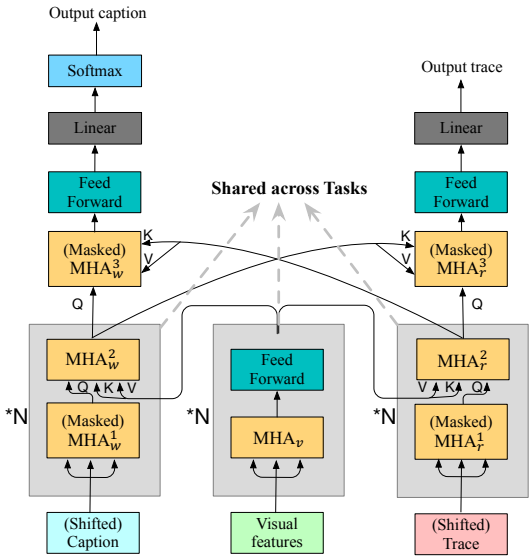

Figure 3: Mirrored Transformer (MITR) architecture. MHA stands for MultiHead Attention.

We then design a mirrored structure for the caption and trace modules, based on the observation that the two modalities are symmetric in the controlled caption generation and controlled trace generation tasks. The caption encoder-decoder, h_w, and trace encoder-decoder, h_r, are defined as:

$$h_w = \text{MultiHead}^2_w(\text{MultiHead}^1_w(x_w, x_w, x_w), h_v, h_v)$$

$$h_r = \text{MultiHead}^2_r(\text{MultiHead}^1_r(x_r, x_r, x_r), h_v, h_v)$$

Our caption and trace modules can switch roles between encoder and decoder seamlessly. Inspired by [26], this switching is implemented by a masking operation, where the encoder observes all inputs but the decoder only observes partial previous information. This prevents the decoder from attending to future information. We implement a masking operation in either MultiHead^1_w or MultiHead^1_r, depending on the specified task:

- For controlled caption generation, the input caption is shifted right by one position, and MultiHead^1_w applies masking to prevent leftward information flow. This ensures every position can only see its previous positions in the attention module. Note, the input trace is not shifted and MultiHead^1_r does not have any masking.
- For controlled trace generation, the input trace is shifted right by one position and MultiHead^1_r applies masking, while the input caption is not shifted and MultiHead^1_w does not perform the mask operation.
- For the joint caption and trace generation task, both the input caption and input trace are shifted right by one position, and both MultiHead^1_w and MultiHead^1_r perform mask operations.

Our model also supports multiple layers. The module between x_v, x_w, x_r and h_v, h_w, h_r can be repeated N times. Specifically, MultiHead_v acts as the encoder, while MultiHead^1_w, MultiHead^2_w, MultiHead^1_r, and MultiHead^2_r switch roles between encoder and decoder depending on what task is being performed. All of these modules are shared across different tasks.

Finally, once h_w and h_r have been computed, MultiHead^3_w and MultiHead^3_r are used to fuse the information from caption and trace modules. Note that in the joint caption and trace generation task, both MultiHead^3_w and MultiHead^3_r need to include a mask operation, while in the other two tasks, no mask operation is needed.

3.3. Controlled Trace Generation: Distance Score

Given a ground truth trace of length q, represented as a sequence of q bounding boxes, and a predicted trace of

length m, we need a score that can measure the distance between these two traces. When $q = m$, the most straightforward way is to compute the $L1$ loss between pairs of bounding boxes (where the two bounding boxes at the same index in the sequence form a pair): $D(\mathbf{r}^{gt}, \hat{\mathbf{r}}) = \frac{1}{q}\sum_{i=1}^{q} |r_i^{gt} - \hat{r_i}|$, where $|r_i^{gt} - \hat{r_i}|$ is the mean $L1$ distance on the four coordinates of the i-th bounding box.

However, there are two main challenges. First, when $q \neq m$, we need to find the exact alignment between the two sets of bounding boxes. Second, even when $q = m$, we may not want to force the two sets to match in the given order because the dataset may contain examples where the local bounding box ordering is not semantically meaningful. [7] shows that if we treat each "trajectory" as a sequence of points on a Riemannian manifold, a distance metric between two trajectories can be derived on a homogeneous space. Such an idea is useful for the case where the sample dimension is much larger than the number of samples. In our case, the sample dimension is small (4-D vector to represent each bounding box), so we choose to simply cast the evaluation task as a simple bipartite matching problem. Note that standard bipartite matching is not a direct solution as it operates on two unordered sets of samples, ignoring the ordering within a trace. Instead, we propose to add local constraints to the bipartite matching so that the orderless matching can only happen within a local window. On the one hand, it provides a way to match two ordered sequences of bounding boxes; on the other hand, it allows local disorder, which is robust to the noise in the dataset annotation.

Consider two traces of lengths q and m; without loss of generality, we assume that $q \leq m$. Let $C \in \mathbb{R}^{q \times m}$ be the cost matrix where C_{ij} is the mean $L1$ distance between the four coordinates of the i-th box from the first trace and j-th box from the second trace, and let X be the assignment matrix. We solve the following linear programming problem to get the distance between these two traces:

$$\begin{aligned} \min_{X} \quad & \mathrm{Tr}(CX^T) \\ s.t., \quad & X\mathbf{1}_m = \mathbf{1}_q, \quad X^T\mathbf{1}_q \leq \mathbf{1}_m, \quad X \geq 0, \quad X_{i,j} = 0, \\ & \forall i, j \quad s.t., \quad 0 \leq i \leq q-1, \quad 0 \leq j \leq m-1, \\ & j < \left\lfloor (i-k)\frac{m}{q} \right\rfloor \quad \text{or} \quad j \geq (i+1+k)\frac{m}{q}, \end{aligned}$$

where $\mathbf{1}_m \in \mathbb{R}^m$ is all one vector and k is the window size controlling the local range of disordered matching. For example, when $q = m$ and $k = 1$, this allows one box from the first trace to match with the box at the same index from the second trace and also its left and right neighbors. After solving this linear programming problem, we use $\mathrm{Tr}(CX^T)/q$ as the distance score between two traces. We call our proposed score the Local Bipartite Matching score (LBM). In addition to using this as an evaluation measure, one could further incorporate this score into training to learn traces for a better matching score, by utilizing recently proposed differentiable linear programming solvers [28, 1]. To keep the presentation succinct, we do not discuss these extensions in this paper.

3.4. Cycle Interaction of Trace and Caption

Another interesting finding of our model architecture is that the controlled trace generation and controlled caption generation are dual problems in one framework, i.e., the output of one direction serves as the input of the other. This inspires us to allow the two modules interact with each other. First, we randomly permute the trace and feed it into the controlled caption generation module, generating the caption. Then, we feed this generated caption[3] into our controlled trace generation model and enforce that the predicted trace be close to the originally permuted trace by adding a cycle loss. By doing so, we enrich the training set by adding more meaningful but unseen trace-caption pairs. As shown in Section 4.5, this further boosts the performance of both tasks.

We denote our mirrored transformer model as $f()$, the controlled trace generation task as $\hat{\mathbf{r}} = f(I, \mathbf{w})$, and the controlled caption generation task as $\hat{\mathbf{w}} = f(I, \mathbf{r})$. We enforce the cycle consistency via

$$L_{\tilde{\mathbf{r}} \to \hat{\mathbf{w}} \to \hat{\mathbf{r}}} = \mathrm{Dist}_{\mathbf{r}}(f(I, f(I, \tilde{\mathbf{r}})), \tilde{\mathbf{r}}),$$

where $\mathrm{Dist}_{\mathbf{r}}$ is the $L1$ loss between the predicted trace and the ground truth trace, and $\tilde{\mathbf{r}}$ is the randomly manipulated trace. Specifically, we perform two types of manipulation: (i) randomly switch the trace within a mini-batch, and (ii) cut a trace into S segments and randomly permute these segments to form a new trace. We show that both manipulations are effective in improving the performance.

3.5. Total Loss Function

The final loss function can be formulated as:

$$\begin{aligned} L_{\text{total}} = \lambda_1 L_{\text{[trace]}} + \lambda_2 L_{\text{[caption]}} \\ + \lambda_3 L_{\tilde{\mathbf{r}} \to \hat{\mathbf{w}} \to \hat{\mathbf{r}}} + \lambda_4 L_{\text{[joint]}}, \end{aligned} \tag{2}$$

where $L_{\text{[trace]}}$ is the $L1$ loss between the predicted trace boxes and ground truth trace boxes for controlled trace generation, $L_{\text{[caption]}}$ is the cross-entropy loss of the caption for controlled caption generation, $L_{\tilde{\mathbf{r}} \to \hat{\mathbf{w}} \to \hat{\mathbf{r}}}$ is the cycle loss, and $L_{\text{[joint]}}$ is the sum of the trace loss and the caption loss for the joint caption and trace generation task.

3.6. Bridging the Gap between Training and Testing

Discrepancies between training and inference always exist in sequential prediction models [5]. At training, the

[3] Gumbel-softmax [18] is applied to approximate the non-differentiable categorical sampling of words.

	# images	# captions	# words/capt
COCO Loc. Narr. [31]	123,287	142,845	41.8
Flickr30k Loc. Narr. [31]	31,783	32,578	57.1
ADE20k Loc. Narr. [31]	22,210	22,529	43.0
Open Images Loc. Narr. [31]	671,469	675,155	34.2

Table 1: Localized Narratives built on top of COCO, Flickr30k, ADE20k, and Open Images.

ground-truth input/output trace at each time step is provided, while at inference the unknown previous trace is replaced by a trace generated by the model itself.

In our proposed joint caption and trace generation task, such discrepancies are even more severe than in standard caption generation, as both the previous word and trace box are generated by the model and thus are connected. The generated trace especially suffers from noise because, unlike the caption, the trace lacks syntax. A single offset could cause the following trace boxes to quickly move to anywhere in the image. To alleviate this problem, we propose a random replacement of the input trace boxes, where we replace a box with $[0, 0, 1, 1, 1]$ (corresponding to the whole image) with probability p. As shown in Table 3, this approach improves the performance of joint caption and trace generation by a clear margin.

4. Experiments

4.1. Dataset

We conduct experiments on four datasets: COCO, Flickr30k, ADE20k, and Open Images, with annotations from two different frameworks: COCO Captions [8] and Localized Narratives [31], summarized in Table 1. We perform an ablation study on COCO and report the performance of our best performing model on the other three datasets in Localized Narratives and the downstream task introduced in Section 4.7 (evaluated on the COCO Captions annotations). We use the COCO2017 split for all experiments except Section 4.7, where we follow [12] to use the split they provide.

The annotations provided by Localized Narratives are challenging to work with for a number of reasons. First, the human-generated trace segment $\mathbf{r}_i$ is a noisy visual representation of the mentioned object, due to imperfect voice-trace synchronization (e.g., if the annotator moves their mouse without speaking), errors in voice-word synchronization (from the automatic sequence-to-sequence alignment model in [31]), inconsistent drawing habits among annotators, and the different nature of mouse trace lines vs. bounding boxes. Second, not every word can be meaningfully grounded in the image, such as existentials (e.g., "there are") and language referring to the observer (e.g., "in this image, I can see ..."). By our estimate, such words account for at least 20% of the words in the COCO validation captions from Localized Narratives. Traces for such words are less meaningful than the other groundable words.

4.2. Experimental Setting

We use our mirrored transformer (MITR) defined in Section 3.2 with $N = 1$ (as shown in Fig. 3). The hidden size of attention layers is 512 and that of the feed-forward layers is 2048. We train the network with batch size 30 using the Adam optimizer [21]. The initial learning rate is $5\text{e}{-4}$, which decays every 3 epochs with decay rate 0.8, for a total of 30 epochs. We use the same training setup for all experiments reported in this paper. The random masking rate for joint caption and trace prediction is $p = 0.5$. In the following, we denote controlled trace generation as Task1, controlled caption generation as Task2, and joint caption and trace generation as Task3.

Controlled Trace Generation In this task, we represent the trace as an ordered sequence of bounding boxes, and the model predicts one bounding box for each word of the input caption, as described in Section 3.1. Given a ground truth trace $\mathbf{r}^{gt} = \{r_1^{gt}, r_2^{gt}, ..., r_T^{gt}\}$ and a predicted trace $\hat{\mathbf{r}} = \{\hat{r}_1, \hat{r}_2, ..., \hat{r}_T\}$ for the same image, we compute the local bipartite matching (LBM) score proposed in Section 3.3 for $k = 0$ and $k = 1$.

Controlled Caption Generation Given an image and a trace, the model predicts the caption corresponding to the trace. When evaluating the quality of generated captions, we report the following widely adopted metrics: BLEU-1, BLEU-4 [29], METEOR [4], ROGUE [23], CIDEr [37], SPICE [2]. We use beam search with size 5.

Joint Caption and Trace Generation In this task, the model is given only an image as input and outputs both caption and trace simultaneously. The model produces outputs iteratively, generating one word and one corresponding bounding box at each time step. At test time, we end the generation when the caption generation branch outputs the `END` token. In this process, since the model itself controls the length of the output, the length of the predicted trace $\hat{\mathbf{r}}$ may differ from the length of the ground truth trace $\mathbf{r}^{gt}$. The max length for generating caption and trace is set to be 100. We report our LBM metric for both $k = 0$ and $k = 1$.

Baselines For controlled trace generation, we construct the baseline by using a standard one-layer encoder-decoder transformer architecture as defined in [36] and feed both visual features and captions to the encoder. Similarly, for controlled caption generation, we use the same architecture as a baseline and feed both visual features and traces to the encoder. For joint caption and trace generation, we construct the baseline by also using the same architecture, but only using visual features as input, and we train it on the caption generation task.

Method	Trained on	BLEU-1	BLEU-4	METEOR	ROUGE$_L$	CIDEr	SPICE
[31]	Task2	0.522	0.246	N/A	0.483	1.065	0.365
Baseline	Task2	0.563	0.255	0.240	0.453	0.997	0.293
MITR	Task2	0.577	0.257	0.245	0.456	1.213	0.293
MITR	Task2 + Task1	0.586	0.272	0.252	0.470	1.329	0.307
MITR	Task2 + Task1 + cycle$_s$	0.596	0.282	0.257	0.476	1.390	0.309
MITR	Task2 + Task1 + cycle$_b$	0.598	0.286	0.258	0.479	1.407	0.313
MITR(2 layer)	Task2 + Task1 + cycle$_b$	**0.607**	**0.292**	**0.263**	**0.487**	**1.485**	**0.317**

Table 2: Quantitative results for Task 2 (controlled caption generation) on COCO. cycle$_s$ and cycle$_b$ refer to two types of cycle loss defined in Sec 4.5. Results from [31] are not directly comparable to ours due to differences mentioned in Sec 4.3.

Method	Trained on	BLEU-1	BLEU-4	METEOR	ROUGE$_L$	CIDEr	SPICE	LBM(k=0)	LBM(k=1)
Baseline	Task2	0.355	0.087	0.155	0.307	**0.310**	0.210	N/A	N/A
MITR	Task3	0.387	0.118	0.168	0.316	0.170	0.194	0.387	0.369
MITR	Task3 + random mask	0.395	**0.128**	**0.184**	**0.328**	0.219	**0.223**	0.308	0.292
MITR	Task3 + Task1 + Task2 + random mask	**0.417**	0.125	0.178	0.323	0.216	0.213	**0.283**	**0.267**

Table 3: Quantitative results for Task 3 (joint caption and trace generation) on COCO.

Dataset	Method	Trained on	BLEU-1	BLEU-4	METEOR	ROUGE$_L$	CIDEr	SPICE	LBM(k=0)	LBM(k=1)
Flickr30k	Baseline	Task1	N/A	N/A	N/A	N/A	N/A	N/A	0.253	0.249
Flickr30k	Baseline	Task2	0.620	0.345	0.286	0.524	1.763	0.341	N/A	N/A
Flickr30k	MITR	Task1 + Task2 + cycle$_b$	**0.644**	**0.374**	**0.300**	**0.547**	**2.014**	**0.365**	**0.195**	**0.188**
ADE20k	Baseline	Task1	N/A	N/A	N/A	N/A	N/A	N/A	0.251	0.247
ADE20k	Baseline	Task2	0.565	0.278	0.259	0.475	1.288	0.341	N/A	N/A
ADE20k	MITR	Task1 + Task2 + cycle$_b$	**0.580**	**0.297**	**0.269**	**0.490**	**1.463**	**0.354**	**0.177**	**0.168**
Open Images	Baseline	Task1	N/A	N/A	N/A	N/A	N/A	N/A	0.212	0.209
Open Images	Baseline	Task2	0.560	0.273	0.261	0.503	1.467	0.361	N/A	N/A
Open Images	MITR	Task1 + Task2 + cycle$_b$	**0.573**	**0.292**	**0.271**	**0.520**	**1.584**	**0.372**	**0.180**	**0.171**

Table 4: Quantitative results for Task 1 and Task2 on Localized Narratives of Flickr30k, ADE20k and Open Images.

4.3. Results on Individual Tasks

Controlled Trace Generation In Fig. 4 (top left), we demonstrate the qualitative results of the controlled trace generation: we can see that the trace closely follows the ground truth trace and also semantically corresponds well to the input caption. Table 5 shows the quantitative results. From the table, we see that our proposed MITR outperforms the baseline method constructed in Section 4.2 (standard transformer with one encoder and one decoder [36]).

Method	Trained on	LBM (k=0)	LBM (k=1)
Baseline	Task1	0.208	0.204
MITR	Task1	0.171	0.159
MITR	Task1 + Task2	0.169	0.157
MITR	Task1 + Task2 + cycle$_s$	0.165	0.156
MITR	Task1 + Task2 + cycle$_b$	0.166	0.155
MITR(2 layer)	Task1 + Task2 + cycle$_b$	**0.163**	**0.154**

Table 5: Quantitative results for Task 1 (controlled trace generation). cycle$_s$ and cycle$_b$ refer to the two cycle losses defined in Sec 4.5. Note: smaller values of LBM are better.

Controlled Caption Generation We show our quantitative results in Table 2 and qualitative results on the right side in Fig. 4. Our baseline model differs from the one in [31] in several places: we use a one-layer encoder-decoder transformer while [31] uses two layers; in addition, we process the trace by cutting the trace by word while [31] cuts the trace by a fixed time interval. Thus, results from [31] are not directly comparable to ours. We mention the performance of [31] in Table 2 for easy reference.

Joint Caption and Trace Generation The quantitative results for this task are in Table 3. We can see that by modeling the trace at the same time as the caption, the performance of caption generation improves by a large margin over the baseline, which only models the caption. In addition, our proposed random masking technique further improves the performance of Task 3 on caption generation by over 1% absolute improvement on all metrics, and on trace generation by nearly 20% relative improvement. The qualitative results are shown in Fig. 4 (bottom). Without human annotated attention traces to guide the caption generation, sometimes the same objects or descriptions are repeated multiple times in a single caption. This suggests that future developments must keep an account of all the objects referenced to avoid repetition.

4.4. Joint Training Results

We demonstrate that, by performing joint training, our model can boost the performance of each individual task

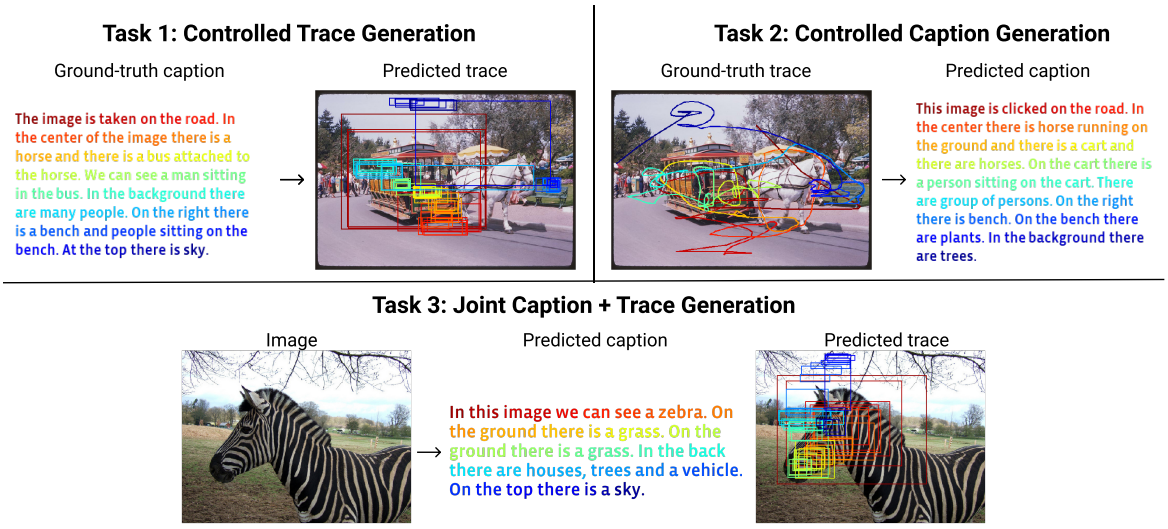

Figure 4: Qualitative results on Tasks 1, 2, and 3 (with more results in the supplementary file).

while using approximately one half of the parameters and training compute cost, compared with training one separate model for each individual task. The quantitative results are in Tables 5 and 2. Further, we can see from Table 3 that the joint training of Task 1 and Task 2 can also help Task 3.

4.5. Cycle Loss Results

We show that, by enforcing cycle consistency, both controlled trace generation and controlled caption generation are further improved when joint training is used. The quantitative results are in Tables 5 and 2. We use cycle_s to represent cycle loss where a single trace is cut into segments and then randomly permuted before forming a new trace, and cycle_b to represent cycle loss where the trace is permuted along the batch dimension within a mini-batch. Adding cycle_b achieves over 1% absolute improvement on BLEU-1 and BLEU-4 compared with our joint training result and over 3% absolute improvement from our baseline model.

4.6. Results on Flickr30k, ADE20k, Open Images

We also report the performance of our best performing model and the baseline model on another three datasets (Flickr30k, AED20k, Open Images), where Localized Narratives are also collected [31]. The results are given in Table 4. As shown, our method achieves consistent improvement over the baseline methods on all datasets.

	B-1	B-4	M	R	C	S
Ours *w/o* pretrain	0.463	0.182	0.219	0.466	1.746	0.363
Ours *w/* pretrain	**0.474**	**0.189**	**0.225**	**0.475**	**1.819**	**0.370**

Table 6: Downstream task on guided caption generation.

4.7. Downstream Task

We further investigate the benefit of our joint training framework. By pre-training using our joint training framework on Localized Narratives [31] and fine-tuning on a guided caption generation task [12] on COCO Captions [8], we are able to get better results than directly training on COCO Captions. In this experiment, we follow [12] to use the COCO split provided by [20].

The task is defined as: given an image I and a sequence of ordered bounding boxes $\mathbf{r} = \{r_1, r_2, \cdots, r_T\}$ as guidance, the model generates a caption $\mathbf{w} = \{w_1, w_2, \cdots, w_N\}$. This task is similar to our controlled caption generation task (Task 2), but we do not assume any correspondence between the boxes and words for both training and testing. Note that [12] considers a slightly different setting, where the dense correspondences between boxes and words are given during training but not at testing. Thus a special gate function was proposed to automatically attend the words to the boxes during test time. See the supplementary material for more details. The results are in Table 6, where pre-training offers a clear gain.

5. Conclusion

We presented a unified framework for modeling vision, language, and human attention traces. Our work is built on top of the Localized Narratives framework and motivated by the need for longform image captions and dense visual grounding. We proposed a Mirrored Transformer model architecture that was jointly trained on three vision-and-language tasks. We demonstrated the effectiveness of our approach through detailed experiments on four datasets.

References

[1] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and Zico Kolter. Differentiable convex optimization layers. *arXiv preprint arXiv:1910.12430*, 2019. 5

[2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: semantic propositional image caption evaluation. *CoRR*, abs/1607.08822, 2016. 6

[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 *CVPR*, 2018. 2, 3

[4] 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*, 2005. 6

[5] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179, 2015. 5

[6] Jeffrey P Bigham, Ryan S Kaminsky, Richard E Ladner, Oscar M Danielsson, and Gordon L Hempton. Webinsight: making web images accessible. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*, 2006. 2

[7] Rudrasis Chakraborty, Vikas Singh, Nagesh Adluru, and Baba C Vemuri. A geometric framework for statistical analysis of trajectories with distinct temporal spans. In *CVPR*, pages 172–181, 2017. 5

[8] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 6, 8

[9] Xinlei Chen and C Lawrence Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *CVPR*, 2015. 2

[10] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Learning universal image-text representations. In *ECCV*, 2020. 1

[11] Maurizio Corbetta and Gordon Shulman. Control of goal-directed and stimulus-driven attention in the brain. *Nature reviews. Neuroscience*, 3:201–15, 04 2002. 2

[12] Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Show, control and tell: A framework for generating controllable and grounded captions. In *CVPR*, 2019. 1, 6, 8

[13] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-Memory Transformer for Image Captioning. In *CVPR*, 2020. 2

[14] Abhishek Das, Harsh Agrawal, Larry Zitnick, Devi Parikh, and Dhruv Batra. Human attention in visual question answering: Do humans and deep networks look at the same regions? *Computer Vision and Image Understanding*, 163:90–100, 2017. 2

[15] Chaorui Deng, Qi Wu, Qingyao Wu, Fuyuan Hu, Fan Lyu, and Mingkui Tan. Visual grounding via accumulated attention. In *CVPR*, 2018. 2

[16] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *ECCV*, 2016. 2

[17] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *ICCV*, 2019. 2

[18] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017. 5

[19] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *CVPR*, 2016. 3

[20] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 2, 8

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[22] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations, 2016. 2

[23] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*. ACL, 2004. 6

[24] Jingyu Liu, Liang Wang, and Ming-Hsuan Yang. Referring expression generation and comprehension via attributes. In *ICCV*, 2017. 3

[25] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, 2019. 1, 2

[26] Lei Zhang Houdong Hu Jason J. Corso Jianfeng Gao Luowei Zhou, Hamid Palangi. Unified vision-language pre-training for image captioning and vqa. In *AAAI*, 2020. 4

[27] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. 2, 3

[28] Zihang Meng, Sathya N Ravi, and Vikas Singh. Physarum powered differentiable linear programming layers and applications. *arXiv preprint arXiv:2004.14539*, 2020. 5

[29] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 6

[30] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015. 2, 3

[31] Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *ECCV*, 2020. 1, 2, 3, 6, 7, 8

[32] Ronald A. Rensink. The dynamic representation of scenes. *Visual Cognition*, 7(1-3):17–42, 2000. 2

[33] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *ECCV*, 2016. 2

[34] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *ICLR*, 2020. 1

[35] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019. 1

[36] 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*, 2017. 3, 6, 7

[37] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 6

[38] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 2

[39] 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 *ICML*, 2015. 2

[40] 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 *ICML*, 2015. 2

[41] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*, 2018. 3

[42] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, 2016. 2, 3

[43] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. A joint speaker-listener-reinforcer model for referring expressions. In *CVPR*, 2017. 3

[44] Yuting Zhang, Luyao Yuan, Yijie Guo, Zhiyuan He, I-An Huang, and Honglak Lee. Discriminative bimodal networks for visual localization and detection with natural language queries. In *CVPR*, 2017. 3