# One Patch to Caption Them All: A Unified Zero-Shot Captioning Model

Lorenzo Bianchi[1,2*] Giacomo Pacini[1,2*] Fabio Carrara[1] Nicola Messina[1]
Giuseppe Amato[1] Fabrizio Falchi[1]
[1]ISTI - CNR, Italy [2]University of Pisa, Italy
name.surname@isti.cnr.it
*Equal contribution

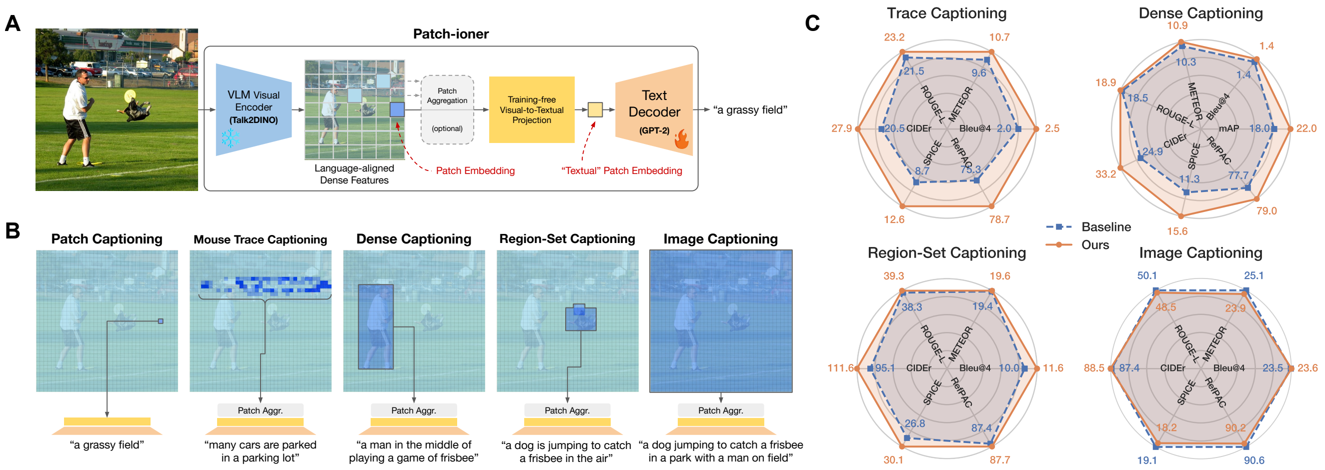


Figure 1. **Overview**. **A**. We present Patch-ioner, a unified zero-shot captioning model based on language-aligned patch representations. We train a text decoder to generate captions from dense representations produced by a Vision Transformer backbone without requiring any image supervision. **B**. By aggregating patch-level representations from different image regions, Patch-ioner can solve several captioning tasks with varying spatial granularities. **C**. We achieve state-of-the-art or competitive performance on various zero-shot region-level captioning tasks, including *zero-shot trace captioning* (in which a mouse trace guides caption control), *dense* and *region-set captioning*.

## Abstract

*We introduce Patch-ioner, a unified zero-shot captioning model that shifts from an image-centric to a patch-centric paradigm, enabling caption generation at arbitrary spatial granularity without region-level supervision. Instead of relying on full-image representations, we treat individual patches as atomic captioning units and aggregate them to describe arbitrary regions, from single patches to non-contiguous areas and entire images. Leveraging language-aligned dense visual representations, we provide a flexible framework for solving various captioning tasks in a zero-shot manner. Experiments demonstrate state-of-the-art performance in zero-shot dense, region-set, and a newly introduced trace captioning task, highlighting the effectiveness of patch-wise semantic representations for scalable caption generation. Website at paciosoft.com/Patch-ioner/.*

## 1. Introduction

Image captioning is one of the most representative tasks in vision-language understanding and reached incredible accuracy thanks to the availability of pre-trained vision-language backbones and large paired image-text datasets. In its basic formulation, a captioning model takes a full image in input and autonomously decides which elements must be described and up to which degree. To enable user guidance and produce more targeted descriptions, some previous works proposed region-level captioning methods [7, 17], which take as an additional input a spatial indication — e.g., bounding boxes — specifying which image regions have to be described and, possibly, in which order.

These region-level captioning methods require an incredible amount of manually crafted data to fully supervise the model. Indeed, each sequence or set of bounding boxes for a given image should correspond to a manually written ground-truth caption describing those objects. Obviously, this fully-supervised solution does not scale properly.

In this paper, we propose a perspective shift that enables us to perform region-level captioning with arbitrary spatial granularity — from a single image patch up to the entire image — in a *zero-shot* fashion, i.e., without relying on a region-annotated dataset. Specifically, instead of relying on the idea that the subject of a captioning method is the *image* — then potentially conditioned on a set of sub-regions — we instead build on two straightforward yet powerful ideas: i) the simplest element that we could caption is a *patch*, the atomic element of an image representation in modern architectures based on vision transformers [10], and ii) we can easily aggregate multiple patches to produce descriptions for arbitrarily large — and also potentially not contiguous — image regions. We call the model implementing these ideas *Patch-ioner*, a zero-shot captioner able to provide a natural language description for single image patches and with the ability to generate a caption for arbitrary aggregation of patches. Our model offers maximum flexibility in zero-shot captioning tasks and can effortlessly generate captions for various aggregations of image patches, ranging from an individual patch to larger image regions, up to providing a caption for the entire image.

Despite the powerful perspective change that defines the patch as the new captioning unit, the problem is now entangled in a simple yet critical question: *how can we craft a model able to provide patch-level captions without relying on any direct patch-level ground truth supervision?*

The astonishing abilities of large pre-trained vision-language foundation models like CLIP [16, 23, 40] have permitted, in the last years, the solving of many downstream tasks in zero-shot or even training-free configurations. In particular, contrastively learned vision-language representations enabled impressive results in zero-shot settings in image classification [40, 50], open-vocabulary detection [30, 53] and segmentation [14, 26], or text-image retrieval [20]. Image captioning, however, cannot directly employ CLIP machinery at inference time to generate text, given that CLIP is inherently a discriminative — and not a generative — approach. Only recently, image captioning models became zero-shot by decoupling image encoding — where pre-trained discriminative models like CLIP are used to create proper image and text representations — from the actual generative module. This is the case for models like [15, 24, 33], which i) employ CLIP to leverage a shared vision-language semantic space, and ii) train a text decoder on solely text samples to recover the text back from the CLIP textual feature. This requires nothing more than a pre-trained contrastive model and a large set of sole text samples to craft a powerful captioner.

Our model follows the same core idea. However, unlike existing zero-shot captioning models, we design it to perform textual decoding on the image patch tokens in output from the underlying vision transformer rather than on the global CLS image representation. Although it may seem an innocuous and straightforward adaptation, the success of this method is bounded by the ability of CLIP to encode fine-grained information in its image patches. Therefore, in this paper, we complete the puzzle by adding two critical components: i) a pre-trained contrastive model which — unlike CLIP — is able to create meaningful patch representations, and ii) ad-hoc aggregation functions that merge the different patch embeddings so that the resulting caption can describe the union of the aggregated patches, without having to re-train the textual decoder.

Thanks to the Patch-ioner and the simple introduced patch-wise aggregation functions, we are able to reach state-of-the-art or comparable results in many zero-shot versions of known captioning task variants — *dense captioning* [17], *region-set captioning* [7], as well as standard image captioning [44] — and in our novel introduced zero-shot *trace captioning* task, which requires to generate a caption for a region within an image specified by a mouse trace.

To summarize, we propose the following contributions:

- We shift perspective for solving an ample set of captioning task variants, transitioning from the widely used *image-to-caption* vision to the *patch-to-caption* approach.
- We introduce the *Patch-ioner*, the first model able to generate captions for single image patches and arbitrary aggregations of them without employing full supervision.
- We probe the Patch-ioner performance on four different image captioning variants, showing the effectiveness of the proposed method despite its overall simplicity.

## 2. Related Work

**Language-aligned Dense Image Representations** are crucial for our goal of captioning at patch level. Vision-language models (VLM) like CLIP [40] introduced a powerful approach to learning global modality representations in a shared space via contrastive learning, paving the way to solve several downstream tasks, including captioning [8, 31]. However, in zero-shot settings, CLIP-like representations are known to struggle with dense tasks due to misalignment between local visual patches and fine-grained semantics [4, 41, 52]. On the other hand, visual-only self-supervised models (SSM) like DINO [5, 35] excel in local semantic modeling but lack a bridge with language. Recent works like SILC [32] and dino.txt [18] attempt to get the best of both worlds by combining DINO- and CLIP-like training objectives, aiming to obtain language-aligned dense representations. Other methods instead exploit already existing VLMs and SSMs to get the same properties with minimal or no training: Talk2DINO [3] connects language to the DINOv2 space by mapping CLIP textual representations to DINOv2 patches. ProxyCLIP [22] instead leverages DINO's attention maps to improve the local properties of the CLIP visual embeddings of patches.

**Zero-shot Image Captioning** methods rely mostly on global CLIP representations to guide text generation. *Early-guided* decoding methods take CLIP visual features as input and introduce adaptation techniques to reduce the visual-textual modality gap [27]. DeCap [24] projects CLIP visual features into a more text-aligned space using a memory of texts as basis, while CapDec [33] and CLOSE [15] inject noise during text-only training to enable decoding also from the CLIP visual space. ViECap [13] enhances captions with entity information extracted via CLIP, and MeaCap [49] refines outputs iteratively by incorporating structured subject-predicate-object knowledge. *Late-guided* decoding methods instead use CLIP as a scoring or optimization signal rather than direct input. ZeroCap [44] leverages CLIP gradients to steer the cached context during text generation, while MAGIC [43] optimizes token selection based on CLIP similarity scores. However, all the above approaches rely on global representations, which are not well-suited for capturing localized semantic details, making them suboptimal for patch-level or region-level captioning in zero-shot settings.

**Region-level Captioning** comprises several tasks in which models are asked to produce natural language descriptions based on sub-parts of an image. They pose additional challenges as naively captioning the cropped regions or feature maps often induces a loss of the global context of the image and, thus, misinterpretation of the region. For this reason, zero-shot solutions to this family of problems are still underexplored. For *controllable captioning* [7] — the generation of an image caption controlled by a set or sequence of regions — and *dense captioning* [17] — the localization and captioning of salient regions of an image — state-of-the-art solutions like CAG-Net [47], GRiT [46], ControlCap [51], and FlexCap [12] provide good performance but need supervision with ground-truth boxes. Below the region-level granularity, we are unaware of existing efforts, but the Localized Narratives dataset [38] — comprising images, timed captions, and timed mouse tracks — provide the ingredients for evaluating captioning also at track- or patch-level. We propose a unique framework to solve captioning at various granularity, from image- to patch-level, in a *zero-shot setting*.

# 3. The *Patch-ioner* Model

We design our *Patch-ioner* model to generate captions for individual patches within an image. To accomplish this, we first obtain language-aligned patch representations in a shared image-text latent space using a transformer-based vision-language model. We then adopt a training-free projection to transform the patch representation (or an aggregation of multiple ones) to mitigate the image-text modality gap. Finally, we adopt a text decoder — trained with a text-only dataset — to generate a caption conditioned on the transformed patch representation. The entire process does not rely on image supervision at any stage.

In §3.1, we describe in detail the caption generation process for a single image patch, which is schematized in Figure 2. In §3.2, we show how our model can solve known region-level and image-level captioning tasks using simple patch aggregation strategies.

## 3.1. Patch-level Captioning

**Patch Feature Extraction.** Let $I \in \mathbb{R}^{H \times W \times 3}$ be an image, which is divided into a grid of non-overlapping patches of size $P \times P$, and $(\psi_\text{v}, \psi_\text{t})$ a pre-trained vision-language model comprised of a transformer-based visual encoder $\psi_\text{v}$ and a textual encoder $\psi_\text{t}$ providing embeddings in a shared space in $\mathbb{R}^D$. We process the image patches and extract a dense feature map $V = \psi_\text{v}(I) \in \mathbb{R}^{\frac{H}{P} \times \frac{W}{P} \times D}$. where each spatial location in $V$ corresponds to the feature representation of a specific patch in the image. Our objective is to generate a caption for the $i$-th patch, utilizing its feature representation $\mathbf{v}_i \in \mathbb{R}^D$ where $i$ indexes a specific patch in $V$. The choice of the vision encoder $\psi_\text{v}$ is crucial, as the quality of patch representations impacts the semantic alignment between image regions and textual descriptions.

**Modality Gap.** Contrastively learned multi-modal spaces suffer from the modality gap phenomenon [27], where image and text embeddings are not perfectly aligned within the shared feature space. To mitigate this issue, we adopt a projection-based decoding mechanism inspired by [24]. Instead of directly using the image patch representation $\mathbf{v}$, we align it with the text embedding space by computing a weighted combination of stored text embeddings from a support memory $M = \{\mathbf{m}_1, \dots, \mathbf{m}_N\}$. Formally,

$$\mathbf{v}_\text{proj} = \sum_{j=1}^{N} \alpha_j \mathbf{m}_j \text{ with } \alpha_j = \frac{\exp((\mathbf{m}_j^\top \mathbf{v})/\tau)}{\sum_{k=1}^{N} \exp((\mathbf{m}_k^\top \mathbf{v})/\tau)} , \tag{1}$$

where

- $\mathbf{m}_i = \psi_\text{t}(t_i)$ represents the text embedding of a sentence $t_i$ belonging to the support memory $M$,
- $\alpha_i$ is the weight assigned to each text embedding, determined by the cosine similarity between the image patch representation $\mathbf{v}$ and each stored text embedding $\mathbf{m}_i$, and
- $\tau$ is a temperature parameter that controls the sharpness of the softmax distribution.

This approach ensures that the patch representation is mapped to a text-aligned embedding. The final projected vector $\mathbf{v}_\text{proj}$ is the input for the text decoder that generates the caption. We also test an alternative solution in SM§7.

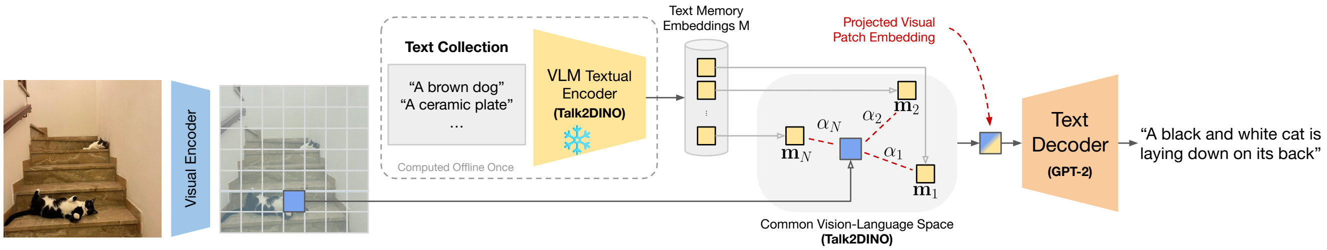


Figure 2. **Patch-level Captioning**. Given an input image, we first extract dense patch-level representations using a vision transformer backbone. For a selected patch, we apply a projection-based decoding mechanism to align its representation with the text embedding space, mitigating the modality gap. Finally, the transformed embedding is fed into a text decoder trained on a text-only corpus, generating a zero-shot caption for the patch.

**Zero-shot Text Decoding.** We train a text decoder $\phi : \mathbb{R}^D \rightarrow \mathcal{T}$ using a prefix language modeling approach, where the decoder learns to generate the caption $t \in \mathcal{T}$ conditioned on its text embedding extracted from the text encoder $\psi_\text{T}(t)$. The decoder is trained solely on text data, using the corresponding text embeddings as prefixes to reconstruct the original sentences. This ensures alignment between the decoder and the text embedding space without requiring image supervision during training. At inference, instead of a text-derived prefix, we use the projected patch embedding $\mathbf{v}_\text{proj}$ as the prefix input to the decoder to generate the patch caption $t$

$$t = \phi(\mathbf{v}_\text{proj})\,. \tag{2}$$

This allows the generation of meaningful captions for image patches in a zero-shot manner, leveraging the learned alignment between patch-level and textual representations.

## 3.2. Region-level Captioning

Building upon patch-level zero-shot captioning, our method can generate captions for specific regions within an image. A region representation is obtained by aggregating the feature embeddings of its constituent patches employing various methodologies that depend on the specific downstream task. Once the aggregated region representation is obtained, we plug it into Eq. 1 and Eq. 2 to obtain a caption for that region. In the following, we detail the specific tasks and associated aggregation functions.

**Image Captioning** involves generating a single caption that describes the entire image. To achieve this, we derive a global representation by aggregating the feature embeddings of all patches within the image. Specifically, given an image $I$, we compute its global representation $\mathbf{v}_I$ as

$$\mathbf{v}_I = \sum_{i=1}^{\frac{HW}{P^2}} w_i \mathbf{v}_i, \tag{3}$$

where $\mathbf{v}_i$ represents the feature embedding of the $i$-th patch, and $w_i$ is the weight assigned to each patch's contribution to the final representation. The weights $w_i$ are chosen employing the criteria explained in §3.3.

**Dense Captioning** requires locating salient regions in an image and generating their descriptions. We focus on the captioning of already defined boxes, effectively removing the localization subtask, which can be tackled using additional region-proposal models. Given a bounding box $B$, we define $S_B$ as the set of patches that intersect with $B$. To obtain the representation $\mathbf{v}_B$ of the region defined by $B$, we aggregate the feature embeddings of the patches in $S_B$

$$\mathbf{v}_B = \sum_{i \in S_B} w_i \mathbf{v}_i\,, \tag{4}$$

where choices of $w_i$ are reported in §3.3.

**Region-set Captioning** consists of generating a single caption for multiple regions within an image, where each region is specified by a distinct bounding box. Given an image $I$ and a set of bounding boxes $\mathfrak{B} = \{B_1, B_2, \dots, B_K\}$, we define $S_{B_k}$ as the set of patches that intersect with the $k$-th bounding box in $\mathfrak{B}$. To represent the entire set of regions, we aggregate the feature embeddings from all selected patches across all bounding boxes, which results in a combined region-level representation

$$\mathbf{v}_\mathfrak{B} = \sum_{i \in \{B_1 \cup \dots \cup B_K\}} \tilde{w}_i \mathbf{v}_i\,, \tag{5}$$

where $\tilde{w}_i$ is the combined and normalized weight of the $i$-th patch considering the contribution of all bounding boxes insisting on that patch:

$$\tilde{w}_i = \frac{\sum_{B \in \mathfrak{B}} [\![ i \in B ]\!] w_i^{(B)}}{\sum_{B \in \mathfrak{B}} \sum_{j \in B} w_j^{(B)}}\,, \tag{6}$$

where $w_j^{(B)}$ is the weight of the $j$-th patch computed for the bounding box $B$ as reported in §3.3, and $[\![\ldots]\!]$ is the indicator function.

**Trace Captioning.** We also define Trace Captioning as generating a caption for a region within an image specified by a mouse trace. This task is particularly useful to obtain localized descriptions of images. For example, consider the understanding of image content by visually impaired users; a trace captioning system can provide not only a description of the main subjects — as image captioners — but also localized descriptions within the picture. Specifically, given an image $I$ and a mouse trace $T = \{p_1, ..., p_L\}$, where $L$ is the number of points in the trace and each point $p_i$ represents its position within the image, we obtain a representation of the traced region by aggregating the patch representations corresponding to the points in $T$. To achieve this, we first identify the sequence of patch indexes $S_T = [i_1, ...i_L]$ that overlap with each trace point and select their corresponding feature representations $\{\mathbf{v}_{i_j}\}_{j=1}^{L}$. We then compute a trace-level representation $\mathbf{v}_T$ by averaging the features of all selected patches

$$\mathbf{v}_T = \sum_{j=1}^{L} w_{i_j} \mathbf{v}_{i_j} \ . \tag{7}$$

The weights $w_{i_j}$ are chosen as explained in §3.3.

## 3.3. Patch Aggregation

In cases where we are not captioning a single patch, we aggregate the selected patches using reasonable criteria. To this aim, we test different aggregation functions for merging the $\mathbf{v}_i$ in the selected set $S$ of visual patches: a) **uniform**, the average box patch representations (i.e., $w = 1/|S|$); b) **gaussian**, for rectangular configurations of contiguous patches — i.e., either the full image or a bounding box; we consider a weighted average of patches representations where central patches weigh more; specifically, we assign to each patch $(a, b)$ coordinates in a uniform square grid $[-1, 1]^2$ (i.e., the top-left and bottom-right patches have $(-1, -1)$ and $(1, 1)$ coordinates, respectively), and weight of $e^{a^2+b^2}$ in the average, and c) **attention**, a weighted average of box patches representations, with patch weights defined as the average attention map of the last layer of $\psi_\text{v}$.

# 4. Experiments

We assess the performance of Patch-ioner on the set of captioning tasks described in §3.2 encompassing a wide spatial granularity range. From local/fine to global/coarse granularity, the tasks are trace captioning, dense captioning, region-set captioning, and image captioning. We evaluate our model in the *zero-shot* setting, as it does not require training with image data.

**Metrics.** All the datasets used to conduct our evaluation provide a textual ground-truth caption for each annotation. Thus, we adopt standard captioning metrics to measure how close the generated captions are to the ground-truth ones. In particular, we report BLEU@4 (B) [36], METEOR (M) [2], ROUGE-L (R) [28], CIDEr (C) [45], SPICE (S) [1] and RefPAC Score (P) [42]. While the first ones are more related to the syntactic similarity between annotations and predictions, the RefPAC Score is a metric that quantifies the distance between two sentences in a semantic way, independently of the terms and phrase structure adopted.

**Architectural Details.** We select Talk2DINO [3] as the vision-language model underlying our patch feature extraction. Talk2DINO adopts as visual encoder $\psi_\text{v}$ the DINOv2 [34] model with registers [9] and textual encoder as $\psi_\text{t}$ the CLIP text encoder augmented with a shallow adapter that maps the space to the DINOv2 one. This provides language-aligned semantically meaningful patch embeddings. For the textual decoder $\phi$, we chose a prefix GPT2-like decoder-only transformer network with 4 attention heads and 4 layers as in [24]. We trained it with a learning rate of $10^{-5}$ with the collection of captions of the COCO training set. We adopted the same collection of 500000 texts as memory bank $M$and set $\tau = 0.01$ in Eq. 1.

**Evaluation Protocol.** We compared our model with available state-of-the-art zero-shot solutions for standard image captioning. We are not aware of any zero-shot solutions for the other region-level tasks. We adapt one of the best image captioning solutions, i.e., DeCap [24], as a strong baseline. We evaluate each task on two datasets — a COCO-derived dataset and an additional dataset such as Visual Genome [21] or Flickr30k [48] — to assess the in-domain and cross-domain performance. Images are fed to every model without cropping. However, we resize them such that they induce the same number of patches for all models. Figure 3 shows qualitative results for each task and model.

## 4.1. Trace Captioning

As introduced in §3, we define Trace Captioning as the generation of a caption for a mouse trace drawn over an image.

**Dataset.** We exploit Localized Narratives [39] — a dataset in which annotators vocally described objects in images while moving the mouse pointer over the described object. The dataset provides temporal annotated voice transcriptions and mouse traces for the images of many standard captioning datasets. We took the labeled COCO [6, 29] and Flickr30K [48] subsets to build the evaluation datasets for the Trace Captioning task. We split long traces and transcriptions for each image into sentences, keeping the traces

|  | COCO |  |  |  |  |  | Flickr30k |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | B | M | R | C | S | P | B | M | R | C | S | P |
| DeCap CLS | 2.0 | 9.6 | 21.5 | 20.5 | 8.7 | 75.3 | 1.2 | 7.7 | 17.7 | 11.2 | 5.6 | 71.0 |
| DeCap Trace | 0.9 | 7.0 | 17.0 | 10.9 | 5.8 | 75.0 | 0.4 | 5.7 | 14.1 | 5.9 | 3.9 | 73.3 |
| Patch-ioner | **2.5** | **10.7** | **23.2** | **27.9** | **12.6** | **78.7** | **1.6** | **9.3** | **19.9** | **18.8** | **9.6** | **77.0** |

Table 1. **Trace Captioning results.**

temporally located between the start and the end of each sentence. We discarded noisy sentences – such as the ones describing image properties (*The image is blurred*, *the image is edited*, ...) and rewrote each sentence removing uncertainties typical of voice descriptions in a more concise and caption-like style through a few-shots prompted LLM (LLama 3 [11]). Further information regarding the generation of the dataset can be found in the supplementary materials. We use the test sets of COCO (Karpathy splits [19]) and Flickr30k. After annotation cleaning, 51 COCO images resulted without clean sentences and were discarded.

**Compared Models.** For Patch-ioner, we use Eq. 7 to aggregate patches underlying a trace. Note that if a patch is crossed multiple times during a trace, it is weighted more. We compare and report the performance of two baseline methods: a) **DeCap CLS**, in which we use the DeCap zero-shot captioner (that works on the visual CLS token of CLIP). This baseline employs the base DeCap model to generate a global image-wise caption and assigns it to all the traces of that image; b) **DeCap Trace**, in which we employ the pretrained image-wise DeCap model, feeding it with the patch token aggregations instead of the original CLS.

## 4.2. Dense Captioning

We assess the performance on dense captioning tasks following the evaluation procedure from [17], omitting the bounding box proposal and evaluating only the bounding box captioning task, using ground-truth boxes as input for the models. In addition to standard caption metrics, for this task, we also report the mAP as originally defined in [17].

**Dataset.** We use the Visual Genome (VG) v1.2 [17, 21] and VG-COCO test splits [25]. The former comprises 5000 images from VG, while the latter contains 2476 images present in both VG and COCO. Both contain multiple bounding box annotations per image with descriptions.

**Compared Models.** We compare with the following baselines adapted from available zero-shot captioning models: a) **DeCap Crop**, in which we apply the DeCap zero-shot captioner to the cropped region defined by the given bounding box; b) **DeCap CLS**, in which we assign to all boxes of an image the caption of the whole image generated by DeCap from the global CLS token, and c) **DeCap Box**, in which we substitute the input CLS token with the same aggregation of patch tokens used in our model.

**Patch Aggregation.** We report only the best-performing aggregation for **Patch-ioner** and **DeCap Box** methods, that are **gaussian** and **uniform**, respectively. However, we noticed that the choice of different aggregations only marginally affects performance in both models (see SM§6).

## 4.3. Region-Set Captioning

Region-set captioning was originally introduced by Cornia et al. [7], and thus, we follow their evaluation protocol.

**Dataset.** We use the Flickr30K Entities [37] and the COCO Entities [7] datasets. Each record comprises an image, a set of bounding boxes of variable length, and a ground-truth controlled caption. We evaluate on the test splits, comprising of images in the Karpathy and Fei-Fei [19] test splits, that consist of 3569 and 1000 images for COCO and Flickr30k versions, respectively.

**Compared Models.** We compare with the following baselines derived in a similar fashion to the dense captioning ones, which are a) **DeCap CLS**, in which we predict the caption of the whole image generated by DeCap from the global CLS token (in fact, ignoring the region set), and b) **DeCap Set**, where we replace the input CLS token with the same aggregation of patch tokens used in our model.

**Patch Aggregation.** We proceed similarly to §4.2. Notice that, in this case, patch weights are computed independently for each box in the set and summed per patch. Therefore, a patch underlying two intersecting boxes will weigh more. Also for this task, we report only the best-performing aggregations (**gaussian** for **Patch-ioner** and **uniform** for **DeCap Set**) and refer the reader to SM§6 for further details.

## 4.4. Image Captioning

We follow the standard evaluation pipeline for zero-shot image captioning, generating captions for the 5000 images in Karpathy's COCO test split. We compare with several state-of-the-art models, that are DeCap [24], CLOSE [15], ZeroCap [44], MAGIC [43], ViECap [13] and CapDec [33].

**Patch Aggregation.** We solve image captioning with our model by aggregating all patch representations extracted by the visual backbone. Among the tested aggregation strategies described in §4.2, we report the best-performing one for **Patch-ioner**, that is **attention**.

|  | Visual Genome (v1.2) |  |  |  |  |  |  | VG-COCO |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | mAP | M | B | R | C | S | P | mAP | M | B | R | C | S | P |
| DeCap CLS | 0.14 | 8.48 | 0.95 | 15.70 | 19.11 | 9.40 | 73.94 | 0.15 | 8.42 | 0.95 | 15.73 | 19.01 | 9.41 | 73.82 |
| DeCap Box | 0.12 | 6.01 | 0.44 | 13.07 | 10.69 | 5.00 | 74.07 | 0.12 | 5.96 | 0.47 | 13.17 | 10.86 | 4.85 | 74.02 |
| DeCap Crop | 0.18 | 10.33 | 1.40 | 18.44 | 24.56 | 11.28 | 77.76 | 0.18 | 10.35 | 1.39 | 18.49 | 24.87 | 11.35 | 77.73 |
| Patch-ioner | **0.22** | **10.82** | **1.43** | **18.82** | **32.80** | **15.48** | **79.14** | **0.22** | **10.89** | **1.45** | **18.88** | **33.19** | **15.63** | **79.05** |

Table 2. **Dense Captioning results.**

| Model | COCO Entities |  |  |  |  |  | Flickr30k Entities |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | B | M | R | C | S | P | B | M | R | C | S | P |
| DeCap Set | 3.5 | 10.6 | 24.0 | 41.6 | 14.1 | 78.8 | 1.0 | 6.1 | 16.3 | 12.9 | 6.4 | 70.4 |
| DeCap CLS | 10.0 | 19.4 | 38.3 | 95.1 | 26.8 | 87.4 | **5.3** | **12.5** | **29.1** | 39.4 | 15.9 | 78.8 |
| Patch-ioner | **11.6** | **19.6** | **39.3** | **111.6** | **30.1** | **87.7** | 5.0 | 12.1 | 28.5 | **43.5** | **16.6** | **78.9** |

Table 3. **Region-Set Captioning results.**

| Model | COCO |  |  |  |  |  | Flickr30k |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | B | M | R | C | S | P | B | M | R | C | S | P |
| ZeroCap [44] | 2.6 | 11.5 | - | 14.6 | 5.5 | - | - | - | - | - | - | - |
| MAGIC [43] | 12.9 | 17.4 | 39.9 | 49.3 | 11.3 | - | 6.2 | 12.2 | 31.3 | 17.5 | 5.9 | - |
| CLOSE [15] | 22.1 | 23.7 | - | 81.2 | 17.7 | - | - | - | - | - | - | - |
| CapDec [33] | 26.4 | 25.1 | **51.8** | 91.8 | - | - | 17.3 | 18.6 | **42.7** | 35.7 | - | - |
| ViECap [13] | **27.2** | 24.8 | - | **92.9** | 18.2 | - | **17.4** | 18.0 | - | 38.4 | 11.2 | - |
| DeCap [24] | 24.7 | 25.0 | - | 91.2 | 18.7 | - | 16.3 | 17.9 | - | 35.7 | 11.1 | - |
| DeCap$^\dagger$ | 23.4 | **25.1** | 50.1 | 87.4 | **19.1** | **90.6** | 15.6 | **18.8** | 42.0 | **40.0** | 12.5 | **84.8** |
| Patch-ioner | 23.6 | 23.9 | 48.5 | 88.5 | 18.2 | 90.2 | 13.7 | 17.1 | 39.5 | 39.3 | 11.5 | 84.2 |

Table 4. **Image Captioning results.**

### 4.5. Results

We report and discuss the results of the compared methods across the four evaluated captioning tasks, from local to global granularity. The results are summarized in Tables 1–4, and qualitative results are visualized in Figure 3.

**Fine-grained Tasks.** In trace and dense captioning tasks, which focus on local visual elements, Patch-ioner outperforms all baselines across all metrics. For trace captioning (Table 1), we significantly improve over DeCap variants, confirming the effectiveness of our patch-based approach for localized descriptions. **DeCap CLS** obtains lower scores as the global caption may not capture the specific content under the trace. Applying DeCap to the trace patches — as our method — generally provides lower performance due to a lack of semantic local knowledge of the CLIP backbone underlying DeCap. Similarly, we achieve substantial gains in all metrics in dense captioning (Table 2). Applying zero-shot captioning to box crops (**DeCap Crop**) produces the strongest baseline, although being the most computationally expensive, while using the caption generated from a global CLIP representation (**DeCap CLS**) or the patch-aggregated one (**DeCap Box**) provides lower scores. We deem existing zero-shot image captioners struggle with fine-grained descriptions, as their captions tend to overlook local visual elements. However, focusing solely on cropped regions disregards the broader scene context, which negatively impacts dense captioning performance. Instead, our method is able to preserve both local details and contextual information and requires only a single visual backbone forward pass per image.

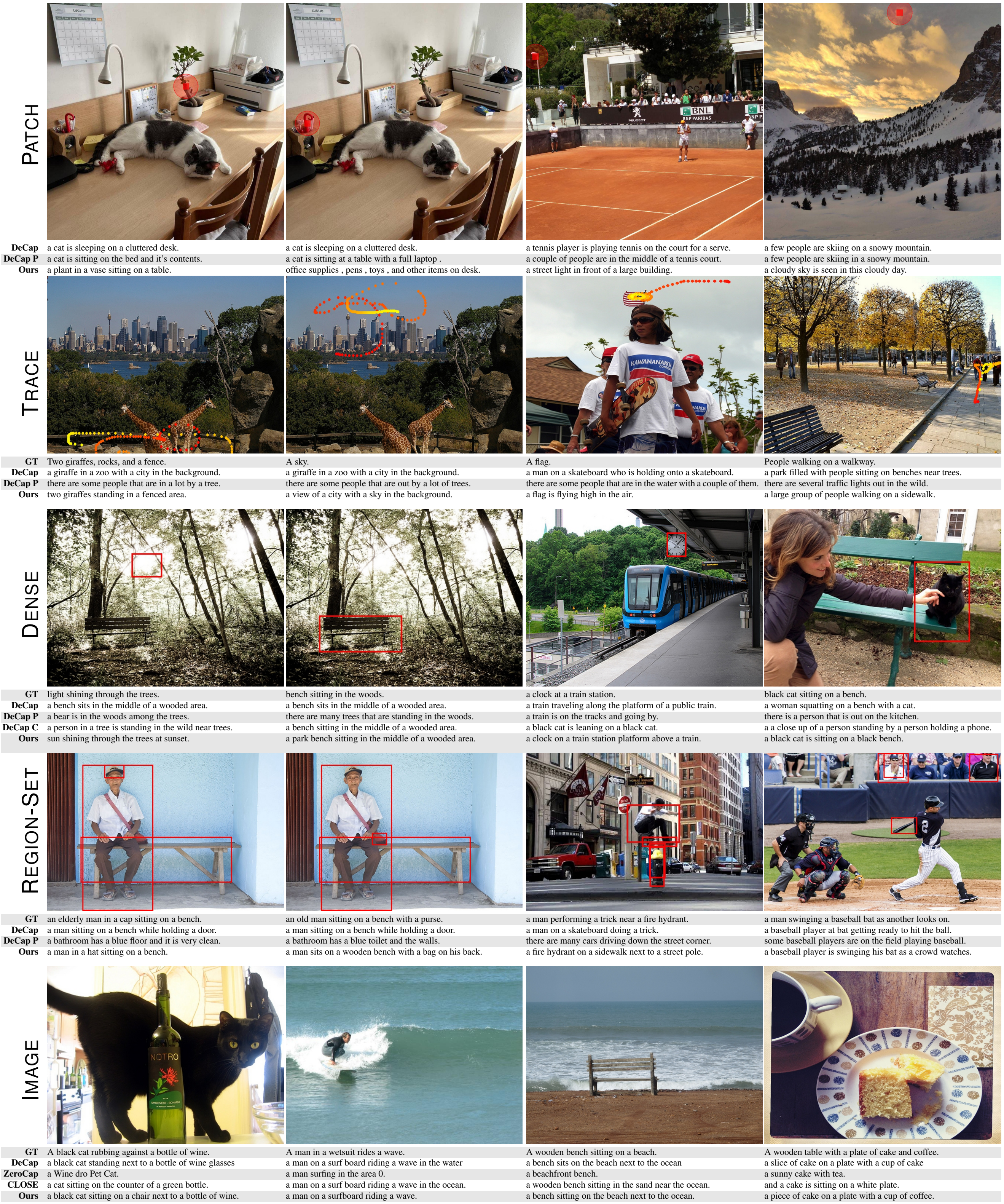
**Context-aware Tasks.** For region-set captioning (Table 3) our model excels on COCO Entities and performs comparably to the best baselines on Flickr30k Entities, with notable improvements in semantic metrics (CIDEr, SPICE, RefPAC). Differently from the previous tasks, we notice the best baseline is the one producing captions for the entire image. This is expected, as the region-set task requires generating a controlled caption of the *whole* image that focuses on the provided regions. We also note a slight performance degradation in the cross-domain setting (Flickr30k), probably due to the decoder bias towards COCO captions. Finally, our method achieves competitive results in standard image captioning (Table 4) compared to state-of-the-art zero-shot captioning models, particularly in the semantic RefPAC Score. The lower scores in syntactic metrics (e.g., BLEU@4) suggest a divergence from human-annotated ground-truth captions in phrasing, yet the semantic quality remains high. This demonstrates the potential of our model to generate meaningful captions without explicit visual-language training also at the image-level.

## 5. Conclusions

In this paper, we introduced the Patch-ioner, a novel zero-shot captioning model that shifts from an image-centric to a patch-centric approach, enabling caption generation for individual patches and arbitrary aggregations without region-level supervision. We rely on the strong spatial awareness of the DINOv2 network, whose local image patches have been effectively bridged with the text modality. Thanks to the disentangled training of the decoder network, this flexible and scalable method enhances various captioning tasks, including dense and region-based captioning, as well as our newly proposed trace captioning.

Despite its simplicity, results show that Patch-ioner can effectively bridge the gap between local and global understanding in image captioning, providing a unified framework for multi-granularity captioning tasks in a zero-shot setting.

In the future, we may enforce an image-level captioning loss to DINO-based contrastively learned representations to obtain better patch-level features in a weakly-supervised manner or enhance patch-to-text feature projection to further reduce the modality gap, critical in zero-shot scenarios.



| | | | | |
|---|---|---|---|---|
| **DeCap** | a cat is sleeping on a cluttered desk. | a cat is sleeping on a cluttered desk. | a tennis player is playing tennis on the court for a serve. | a few people are skiing on a snowy mountain. |
| **DeCap P** | a cat is sitting on the bed and it's contents. | a cat is sitting at a table with a full laptop . | a couple of people are in the middle of a tennis court. | a few people are skiing in a snowy mountain. |
| **Ours** | a plant in a vase sitting on a table. | office supplies , pens , toys , and other items on desk. | a street light in front of a large building. | a cloudy sky is seen in this cloudy day. |

| | | | | |
|---|---|---|---|---|
| **GT** | Two giraffes, rocks, and a fence. | A sky. | A flag. | People walking on a walkway. |
| **DeCap** | a giraffe in a zoo with a city in the background. | a giraffe in a zoo with a city in the background. | a man on a skateboard who is holding onto a skateboard. | a park filled with people sitting on benches near trees. |
| **DeCap P** | there are some people that are in a lot by a tree. | there are some people that are out by a lot of trees. | there are some people that are in the water with a couple of them. | there are several traffic lights out in the wild. |
| **Ours** | two giraffes standing in a fenced area. | a view of a city with a sky in the background. | a flag is flying high in the air. | a large group of people walking on a sidewalk. |

| | | | | |
|---|---|---|---|---|
| **GT** | light shining through the trees. | bench sitting in the woods. | a clock at a train station. | black cat sitting on a bench. |
| **DeCap** | a bench sits in the middle of a wooded area. | a bench sits in the middle of a wooded area. | a train traveling along the platform of a public train. | a woman squatting on a bench with a cat. |
| **DeCap P** | a bear is in the woods among the trees. | there are many trees that are standing in the woods. | a train is on the tracks and going by. | there is a person that is out on the kitchen. |
| **DeCap C** | a person in a tree is standing in the wild near trees. | a bench sitting in the middle of a wooded area. | a black cat is leaning on a black cat. | a a close up of a person standing by a person holding a phone. |
| **Ours** | sun shining through the trees at sunset. | a park bench sitting in the middle of a wooded area. | a clock on a train station platform above a train. | a black cat is sitting on a black bench. |

| | | | | |
|---|---|---|---|---|
| **GT** | an elderly man in a cap sitting on a bench. | an old man sitting on a bench with a purse. | a man performing a trick near a fire hydrant. | a man swinging a baseball bat as another looks on. |
| **DeCap** | a man sitting on a bench while holding a door. | a man sitting on a bench while holding a door. | a man on a skateboard doing a trick. | a baseball player at bat getting ready to hit the ball. |
| **DeCap P** | a bathroom has a blue floor and it is very clean. | a bathroom has a blue toilet and the walls. | there are many cars driving down the street corner. | some baseball players are on the field playing baseball. |
| **Ours** | a man in a hat sitting on a bench. | a man sits on a wooden bench with a bag on his back. | a fire hydrant on a sidewalk next to a street pole. | a baseball player is swinging his bat as a crowd watches. |

| | | | | |
|---|---|---|---|---|
| **GT** | A black cat rubbing against a bottle of wine. | A man in a wetsuit rides a wave. | A wooden bench sitting on a beach. | A wooden table with a plate of cake and coffee. |
| **DeCap** | a black cat standing next to a bottle of wine glasses | a man on a surf board riding a wave in the water | a bench sits on the beach next to the ocean | a slice of cake on a plate with a cup of cake |
| **ZeroCap** | a Wine dro Pet Cat. | a man surfing in the area 0. | a beachfront bench. | a sunny cake with tea. |
| **CLOSE** | a cat sitting on the counter of a green bottle. | a man on a surf board riding a wave in the ocean. | a wooden bench sitting in the sand near the ocean. | and a cake is sitting on a white plate. |
| **Ours** | a black cat sitting on a chair next to a bottle of wine. | a man on a surfboard riding a wave. | a bench sitting on the beach next to the ocean. | a piece of cake on a plate with a cup of coffee. |

Figure 3. **Qualitative results.** We report four predictions of our model and compare baselines from the finer (top) to the coarser (bottom) task. For trace captioning examples, the trace time is color-coded from start (red) to end (yellow). **DeCap** = DeCap applied on the whole image. **DeCap P** = DeCap applied on the same aggregation of patches used by our method. **DeCap C** = DeCap applied on cropped box. **ZeroCap** = ZeroCap [44] applied to the whole image. **CLOSE** = CLOSE [15] applied to the whole image. **GT** = ground-truth caption.

# References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 382–398. Springer, 2016. 5

[2] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. 5

[3] Luca Barsellotti, Lorenzo Bianchi, Nicola Messina, Fabio Carrara, Marcella Cornia, Lorenzo Baraldi, Fabrizio Falchi, and Rita Cucchiara. Talking to dino: Bridging self-supervised vision backbones with language for open-vocabulary segmentation. *arXiv preprint arXiv:2411.19331*, 2024. 2, 5, 1, 3

[4] Ioana Bica, Anastasija Ilic, Matthias Bauer, Goker Erdogan, Matko Bošnjak, Christos Kaplanis, Alexey A Gritsenko, Matthias Minderer, Charles Blundell, Razvan Pascanu, et al. Improving fine-grained understanding in image-text pre-training. In *ICML*, 2024. 2

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *ICCV*, 2021. 2

[6] 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. 5

[7] Marcella Cornia, Lorenzo Baraldi, and Rita 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. 1, 2, 3, 6, 5

[8] Marcella Cornia, Lorenzo Baraldi, Giuseppe Fiameni, and Rita Cucchiara. Generating more pertinent captions by leveraging semantics and style on multi-source datasets. *International Journal of Computer Vision*, 132(5):1701–1720, 2024. 2

[9] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision Transformers Need Registers. In *ICLR*, 2024. 5

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021. 2

[11] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 6

[12] Debidatta Dwibedi, Vidhi Jain, Jonathan J Tompson, Andrew Zisserman, and Yusuf Aytar. Flexcap: Describe anything in images in controllable detail. *Advances in Neural Information Processing Systems*, 37:111172–111198, 2025. 3

[13] Junjie Fei, Teng Wang, Jinrui Zhang, Zhenyu He, Chengjie Wang, and Feng Zheng. Transferable decoding with visual entities for zero-shot image captioning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3136–3146, 2023. 3, 6, 7

[14] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *ECCV*, 2022. 2

[15] Sophia Gu, Christopher Clark, and Aniruddha Kembhavi. I can't believe there's no images! learning visual tasks using only language supervision. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2672–2683, 2023. 2, 3, 6, 7, 9, 1, 8

[16] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *ICML*, 2021. 2

[17] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *CVPR*, pages 4565–4574, 2016. 1, 2, 3, 6

[18] Cijo Jose, Théo Moutakanni, Dahyun Kang, Federico Baldassarre, Timothée Darcet, Hu Xu, Daniel Li, Marc Szafraniec, Michaël Ramamonjisoa, Maxime Oquab, et al. Dinov2 meets text: A unified framework for image- and pixel-level vision-language alignment. *arXiv preprint arXiv:2412.16334*, 2024. 2

[19] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 6

[20] Giorgos Kordopatis-Zilos, Vladan Stojnić, Anna Manko, Pavel Šuma, Nikolaos-Antonios Ypsilantis, Nikos Efthymiadis, Zakaria Laskar, Jiří Matas, Ondřej Chum, and Giorgos Tolias. ILIAS: Instance-level image retrieval at scale, 2025. 2

[21] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017. 5, 6

[22] Mengcheng Lan, Chaofeng Chen, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. ProxyCLIP: Proxy Attention Improves CLIP for Open-Vocabulary Segmentation. In *ECCV*, 2024. 2, 3

[23] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *ICML*, 2023. 2

[24] Wei Li, Linchao Zhu, Longyin Wen, and Yi Yang. Decap: Decoding CLIP latents for zero-shot captioning via text-only training. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. 2, 3, 5, 6, 7, 1

[25] Xiangyang Li, Shuqiang Jiang, and Jungong Han. Learning object context for dense captioning. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8650–8657, 2019. 6

[26] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *CVPR*, 2023. 2

[27] Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *Advances in Neural Information Processing Systems*, 35:17612–17625, 2022. 3

[28] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, 2004. Association for Computational Linguistics. 5

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014. 5

[30] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36:72983–73007, 2023. 2

[31] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 2

[32] Muhammad Ferjad Naeem, Yongqin Xian, Xiaohua Zhai, Lukas Hoyer, Luc Van Gool, and Federico Tombari. SILC: Improving Vision Language Pretraining with Self-Distillation. In *ECCV*, 2024. 2

[33] David Nukrai, Ron Mokady, and Amir Globerson. Text-only training for image captioning using noise-injected CLIP. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 4055–4063. Association for Computational Linguistics, 2022. 2, 3, 6, 7, 1

[34] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning Robust Visual Features without Supervision. *arXiv preprint arXiv:2304.07193*, 2023. 5, 1

[35] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024, 2024. 2

[36] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002. 5

[37] 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 *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015. 6

[38] Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 647–664. Springer, 2020. 3, 4

[39] Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 647–664. Springer, 2020. 5

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 2021. 2

[41] Kanchana Ranasinghe, Brandon McKinzie, Sachin Ravi, Yinfei Yang, Alexander Toshev, and Jonathon Shlens. Perceptual Grouping in Contrastive Vision-Language Models. In *ICCV*, 2023. 2

[42] Sara Sarto, Manuele Barraco, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Positive-augmented contrastive learning for image and video captioning evaluation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6914–6924, 2023. 5

[43] Yixuan Su, Tian Lan, Yahui Liu, Fangyu Liu, Dani Yogatama, Yan Wang, Lingpeng Kong, and Nigel Collier. Language models can see: Plugging visual controls in text generation. *arXiv preprint arXiv:2205.02655*, 2022. 3, 6, 7

[44] Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zerocap: Zero-shot image-to-text generation for visual-semantic arithmetic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17918–17928, 2022. 2, 3, 6, 7, 9, 8

[45] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. 5

[46] Jialian Wu, Jianfeng Wang, Zhengyuan Yang, Zhe Gan, Zicheng Liu, Junsong Yuan, and Lijuan Wang. Grit: A generative region-to-text transformer for object understanding. In *European Conference on Computer Vision*, pages 207–224. Springer, 2024. 3

[47] Guojun Yin, Lu Sheng, Bin Liu, Nenghai Yu, Xiaogang Wang, and Jing Shao. Context and attribute grounded dense captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6241–6250, 2019. 3

[48] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New

similarity metrics for semantic inference over event descriptions. *Transactions of the association for computational linguistics*, 2:67–78, 2014. 5

[49] Zequn Zeng, Yan Xie, Hao Zhang, Chiyu Chen, Bo Chen, and Zhengjue Wang. Meacap: Memory-augmented zero-shot image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14100–14110, 2024. 3

[50] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18123–18133, 2022. 2

[51] Yuzhong Zhao, Yue Liu, Zonghao Guo, Weijia Wu, Chen Gong, Qixiang Ye, and Fang Wan. Controlcap: Controllable region-level captioning. In *European Conference on Computer Vision*, pages 21–38. Springer, 2024. 3

[52] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. RegionCLIP: Region-Based Language-Image Pretraining. In *CVPR*, 2022. 2

[53] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting Twenty-Thousand Classes Using Image-Level Supervision. In *ECCV*, 2022. 2

# One Patch to Caption Them All: A Unified Zero-Shot Captioning Model

Supplementary Material

## 6. Additional Results: Weighting Strategies, Input Resolution, Text Collection

We tested several patch weighting strategies (described in §3.3) and input resolutions for our model and the DeCap-based baselines.

For Patch-ioner, we followed Talk2DINO [3] and used an input image resolution of 518x518, obtaining 37 14x14 patches per side. The original DeCap implementation [24] uses the CLIP B/16 backbone with 224x224 input images with 14 patches per side. We also tested DeCap with a 592x592 input image size in order to obtain the same number of patches of our model (37 per side) and DeCap with the B/32 backbone – the original configuration realeased by [24] –, which results in 7 patches per side.

While the main paper reports only the best configuration per task and model, in this section, we report and discuss the results of all the tested configurations. We perform these tests on the COCO-derived datasets and on VG v1.2 for dense captioning. We highlight the rows in the tables corresponding to the configurations that have been reported in the main paper.

**Trace Captioning.** Table 5 reports trace captioning results from which we conclude the simple average of trace patches provides the best performance. For this task, we did not apply the *gaussian* weighting scheme, as we could not clearly identify rectangular regions from the sparse discontinuous traces.

**Dense Captioning.** Table 6 reports the results of the dense captioning task. For Patch-ioner, changing the weighting strategy does not cause significant performance changes. We observe that DeCap applied to patches is more effective when the number of patches is higher. Indeed, DeCap@592 CLIP B16 — the one having the same number of patches per image as our method — achieves the highest semantic score (in terms of RefPAC-S) among the baselines applied to patches. The best configuration for DeCap is to apply it to CLS tokens of box crops (DeCap@224 Crop). The gap between the CLIP B/16 and B/32 versions is negligible.

**Region-Set Captioning.** Tables 7 and 8 show the results in the region-set captioning task on COCO and Flickr30k, respectively. As already discussed in §4, the baseline methods applied to patches achieve lower scores compared to the one applied to the global CLS token. This is due to the more global nature of the task, which requires the model to produce a caption for the whole image while focusing on certain regions. Examining the results of our Patch-ioner, we notice a reversal in the ranking of configurations between the two datasets. Also, in this task, the weighting strategy only marginally affects performance.

**Image Captioning.** In Table 9, we report the results of standard zero-shot image captioning. In addition to the already described weighting schemes, we test two additional configurations for Patch-ioner that are a) *central patch*, where the decoding is applied to the central patch of the image, and b) *CLS*, where we decode the CLS token provided by the Talk2DINO visual backbone. We can observe that the most effective strategy for the image captioning task is *attention*. This is coherent with results from [3], where they suggest the attention-weighted patch means to use Talk2DINO for global tasks such as image-text retrieval.

**Memory Bank.** Considering that we tackle the modality gap through a projection based on a collection of texts, we tested how much the performance is influenced by the selection of the texts in the memory bank. In Table 9, we also report the results obtained by two of the best configurations of our method when in its memory bank, there are also the ground-truth captions from the test set (rows marked with *GT Memory*). We can observe only a slight performance improvement obtained through this tweak.

## 7. Modality Gap: Projection to Textual Space vs Training with Noise

In this section, we quantitatively assess the performance of two state-of-the-art solutions to overcome the modality gap. In particular, we compared the configuration based on a memory bank of texts — the one introduced in §3 — with an alternative solution based on noise injection during the decoder training.

**Training with Noise.** Various works [15, 33] proposed zero-shot image captioning solutions based on noise injection during the training of the text decoder. Through this strategy, the trained decoders are more effective in understanding semantic representations, even when those are not from texts. To implement this strategy in our framework, we trained the textual decoder on the same collection of captions as for the memory bank-based configuration. We adopted as textual space for the decoder the one of Talk2DINO [3], which is aligned to DINOv2 [34]

| Model | # Patches | Backbone | Input | Weighting | B | M | R | C | S | P |
|---|---|---|---|---|---|---|---|---|---|---|
| DeCap@592 | 37 | CLIP B16 | CLS | - | 2.0 | 9.6 | 21.5 | 20.5 | 8.7 | 75.3 |
| DeCap@592 | 37 | CLIP B16 | Patches | uniform | 0.9 | 7.0 | 17.0 | 10.9 | 5.8 | 75.0 |
| DeCap@592 | 37 | CLIP B16 | Patches | attention | 0.8 | 6.5 | 15.1 | 9.4 | 5.6 | 74.9 |
| Patch-ioner@518 | 37 | DINOv2 B14 | CLS | - | 1.8 | 8.7 | 20.6 | 20.5 | 8.7 | 75.0 |
| Patch-ioner@518 | 37 | DINOv2 B14 | Patches | uniform | **2.5** | **10.7** | **23.2** | **27.9** | **12.6** | **78.7** |
| Patch-ioner@518 | 37 | DINOv2 B14 | Patches | attention | 2.4 | 10.4 | 22.7 | 27.6 | 12.0 | 78.1 |

Table 5. Trace Captioning results on COCO test set.

| Model | # Patches | Backbone | Input | Weighting | mAP | M | B | R | C | S | P |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DeCap@224 | 7 | CLIP B32 | CLS | - | 0.15 | 8.40 | 0.94 | 15.61 | 19.38 | 9.38 | 73.71 |
| DeCap@224 | 7 | CLIP B32 | Patches | uniform | 0.09 | 4.44 | 0.19 | 11.03 | 5.39 | 2.29 | 72.03 |
| DeCap@224 | 7 | CLIP B32 | Patches | gaussian | 0.09 | 4.44 | 0.19 | 11.06 | 5.41 | 2.27 | 71.90 |
| DeCap@224 | 7 | CLIP B32 | Patches | attention | 0.10 | 4.54 | 0.20 | 11.32 | 5.70 | 2.43 | 71.71 |
| DeCap@224 | 14 | CLIP B16 | CLS | - | 0.14 | 8.48 | 0.95 | 15.70 | 19.11 | 9.40 | 73.94 |
| DeCap@224 | 14 | CLIP B16 | Patches | uniform | 0.12 | 6.01 | 0.44 | 13.07 | 10.69 | 5.00 | 74.07 |
| DeCap@224 | 14 | CLIP B16 | Patches | gaussian | 0.12 | 5.97 | 0.45 | 13.06 | 10.58 | 4.89 | 73.91 |
| DeCap@224 | 14 | CLIP B16 | Patches | attention | 0.13 | 5.91 | 0.43 | 12.95 | 10.33 | 4.84 | 73.73 |
| DeCap@592 | 37 | CLIP B16 | CLS | - | 0.15 | 8.37 | 0.92 | 15.67 | 18.53 | 9.26 | 73.91 |
| DeCap@592 | 37 | CLIP B16 | Patches | uniform | 0.12 | 5.97 | 0.45 | 13.15 | 10.89 | 4.87 | 74.15 |
| DeCap@592 | 37 | CLIP B16 | Patches | gaussian | 0.12 | 5.93 | 0.44 | 13.14 | 10.74 | 4.76 | 73.99 |
| DeCap@592 | 37 | CLIP B16 | Patches | attention | 0.12 | 5.80 | 0.42 | 12.89 | 10.22 | 4.66 | 73.89 |
| DeCap@224 Crop | 7 | CLIP B32 | CLS | - | 0.17 | 10.03 | 1.35 | 18.20 | 23.61 | 10.90 | 77.09 |
| DeCap@224 Crop | 14 | CLIP B16 | CLS | - | 0.18 | 10.33 | 1.40 | 18.44 | 24.56 | 11.28 | 77.76 |
| Patch-ioner@518 | 37 | DINOv2 B14 | Patches | uniform | 0.21 | 10.63 | 1.36 | 18.59 | 31.94 | 15.03 | 78.82 |
| Patch-ioner@518 | 37 | DINOv2 B14 | Patches | gaussian | **0.22** | **10.82** | **1.43** | **18.82** | **32.80** | **15.48** | **79.14** |
| Patch-ioner@518 | 37 | DINOv2 B14 | Patches | attention | 0.21 | 10.31 | 1.27 | 18.17 | 30.58 | 14.72 | 78.69 |

Table 6. Dense Captioning results on VG v1.2 test set.

with registers [18]. Following the setting of [15], we added Gaussian noise with $\sigma^2 = 0.08$ to the textual embeddings while leaving the other parameters unchanged. In the next paragraphs, we report and compare the results for each task of the Patch-ioner model that uses the memory bank ($\mathcal{M}emory$) and the one trained with noise ($\mathcal{N}oise$).

In Table 10, we compare the two modality gap mitigation strategies across multiple captioning tasks. For Trace Captioning (Table 10a), the $\mathcal{M}emory$ method is slightly more effective in the semantic metric RefPAC-S, while the $\mathcal{N}oise$ variant achieves marginally better scores in CIDEr, ROUGE-L, METEOR, and BLEU@4, with a minimal gap between the two approaches. In Dense Captioning (Table 10b), the $\mathcal{M}emory$ model consistently outperforms the $\mathcal{N}oise$ model across all metrics. Similarly, for Region-Set Captioning (Table 10c), both methods achieve strong results, but the $\mathcal{M}emory$ method shows a clearer advantage, particularly in tasks closer to the patch level. Finally, in Image Captioning (Tables 10d and 10e), the performance gap between the two architectures narrows, especially on the Flickr30k test split. In this scenario, the $\mathcal{M}emory$ method performs significantly better when applied to the CLS token, whereas patch aggregation produces comparable results. However, the metrics reveal conflicting trends across different datasets.

**Chosen Strategy.** Based on the observed results, we selected the projection-based approach ($\mathcal{M}emory$) as the primary strategy for overcoming the modality gap in our framework. While the noise injection method ($\mathcal{N}oise$) yielded competitive performance across multiple tasks, the $\mathcal{M}emory$ method demonstrated superior performance in dense captioning and region-set captioning, as well as a clear advantage when applied to the CLS token in image captioning. Given these trends, and considering the stability of the projection-based approach across different evaluation settings, we adopted $\mathcal{M}emory$ as the default configuration for our Patch-ioner.

| Model | # Patches | Backbone | Input | Weighting | B | M | R | C | S | P |
|---|---|---|---|---|---|---|---|---|---|---|
| DeCap@224 | 7 | CLIP B32 | CLS | - | 10.1 | 19.0 | 38.0 | 94.4 | 26.4 | 86.9 |
| DeCap@224 | 7 | CLIP B32 | Patches | uniform | 1.1 | 6.6 | 17.3 | 15.8 | 5.8 | 71.2 |
| DeCap@224 | 7 | CLIP B32 | Patches | gaussian | 1.1 | 6.5 | 17.2 | 15.3 | 5.6 | 71.0 |
| DeCap@224 | 7 | CLIP B32 | Patches | attention | 1.1 | 6.5 | 17.3 | 15.6 | 5.6 | 70.8 |
| DeCap@224 | 14 | CLIP B16 | CLS | - | 10.0 | 19.4 | 38.3 | 95.1 | 26.8 | 87.4 |
| DeCap@224 | 14 | CLIP B16 | Patches | uniform | 3.1 | 10.5 | 23.6 | 40.8 | 14.4 | 78.7 |
| DeCap@224 | 14 | CLIP B16 | Patches | gaussian | 3.1 | 10.5 | 23.4 | 40.1 | 14.1 | 78.5 |
| DeCap@224 | 14 | CLIP B16 | Patches | attention | 2.5 | 9.4 | 21.4 | 34.2 | 12.8 | 77.3 |
| DeCap@592 | 37 | CLIP B16 | CLS | - | 9.6 | 18.6 | 37.5 | 91.4 | 25.9 | 86.7 |
| DeCap@592 | 37 | CLIP B16 | Patches | uniform | 3.5 | 10.6 | 24.0 | 41.6 | 14.1 | 78.8 |
| DeCap@592 | 37 | CLIP B16 | Patches | gaussian | 3.5 | 10.5 | 23.8 | 40.6 | 13.8 | 78.5 |
| DeCap@592 | 37 | CLIP B16 | Patches | attention | 2.8 | 9.2 | 21.4 | 33.6 | 12.4 | 77.1 |
| Patch-ioner@518 | 37 | DINOv2 B14 | CLS | - | 9.1 | 16.9 | 35.0 | 89.4 | 25.4 | 85.5 |
| Patch-ioner@518 | 37 | DINOv2 B14 | Patches | uniform | 11.5 | 19.3 | 38.8 | 109.1 | 29.4 | 87.5 |
| Patch-ioner@518 | 37 | DINOv2 B14 | Patches | gaussian | **11.6** | **19.6** | **39.3** | **111.6** | **30.1** | **87.7** |
| Patch-ioner@518 | 37 | DINOv2 B14 | Patches | attention | 11.0 | 19.0 | 38.3 | 107.0 | 29.3 | 87.4 |

Table 7. Region-Set Captioning results for COCO Entities test set.

| Model | # Patches | Backbone | Input | Weighting | B | M | R | C | S | P |
|---|---|---|---|---|---|---|---|---|---|---|
| DeCap@224 | 7 | CLIP B32 | CLS | - | 5.1 | 12.0 | 28.6 | 37.5 | 14.7 | 78.2 |
| DeCap@224 | 7 | CLIP B32 | Patches | uniform | 0.5 | 4.8 | 14.1 | 5.9 | 3.3 | 65.3 |
| DeCap@224 | 7 | CLIP B32 | Patches | gaussian | 0.5 | 4.8 | 14.1 | 5.9 | 3.2 | 65.2 |
| DeCap@224 | 7 | CLIP B32 | Patches | attention | 0.5 | 5.0 | 14.6 | 6.2 | 3.4 | 65.2 |
| DeCap@224 | 14 | CLIP B16 | CLS | - | **5.3** | **12.5** | **29.1** | 39.4 | 15.9 | 78.8 |
| DeCap@224 | 14 | CLIP B16 | Patches | uniform | 1.0 | 5.9 | 15.6 | 12.4 | 6.7 | 70.4 |
| DeCap@224 | 14 | CLIP B16 | Patches | gaussian | 0.9 | 5.9 | 15.5 | 12.6 | 6.6 | 70.4 |
| DeCap@224 | 14 | CLIP B16 | Patches | attention | 1.0 | 5.6 | 14.6 | 11.8 | 6.1 | 69.9 |
| DeCap@592 | 37 | CLIP B16 | CLS | - | 5.0 | 12.0 | 28.5 | 37.8 | 14.9 | 78.1 |
| DeCap@592 | 37 | CLIP B16 | Patches | uniform | 1.0 | 6.1 | 16.3 | 12.9 | 6.4 | 70.4 |
| DeCap@592 | 37 | CLIP B16 | Patches | gaussian | 1.0 | 6.1 | 16.4 | 12.9 | 6.3 | 70.4 |
| DeCap@592 | 37 | CLIP B16 | Patches | attention | 0.9 | 5.6 | 15.0 | 11.5 | 5.8 | 69.8 |
| Patch-ioner@518 | 37 | DINOv2 B14 | CLS | - | 3.8 | 10.5 | 26.3 | 35.7 | 13.4 | 76.6 |
| Patch-ioner@518 | 37 | DINOv2 B14 | Patches | gaussian | 5.0 | 12.1 | 28.5 | **43.5** | **16.6** | **78.9** |
| Patch-ioner@518 | 37 | DINOv2 B14 | Patches | uniform | 5.1 | 12.0 | 28.7 | 44.1 | 16.4 | 79.1 |
| Patch-ioner@518 | 37 | DINOv2 B14 | Patches | attention | 4.8 | 11.6 | 27.8 | 41.9 | 15.8 | 78.6 |

Table 8. Region-Set Captioning results on Flickr30k Entities test set.

## 8. Dense Vision-Language Backbone

In this section, we explore alternative approaches to Talk2DINO to obtain language-aligned dense representation. As outlined in §3, our approach relies on a vision-language backbone that provides localized dense features. We experimented with the two best available backbones: ProxyCLIP [22] and Talk2DINO [3], with Talk2DINO being the preferred choice for Patch-ioner. Both backbones were proposed to address the task of unsupervised open-vocabulary object segmentation, in which Talk2DINO emerges as the best-performing model.

ProxyCLIP leverages the strong semantic capabilities of DINO model families to generate dense CLIP features,

| Model | # Patches | Backbone | Input | Weighting | B | M | R | C | S | P |
|---|---|---|---|---|---|---|---|---|---|---|
| DeCap@224 | 14 | CLIP B16 | CLS | - | **23.89** | **25.51** | **50.34** | **89.64** | **19.52** | **91.05** |
| DeCap@224 | 14 | CLIP B16 | Patches | uniform | 9.05 | 15.76 | 33.74 | 41.68 | 10.72 | 84.15 |
| DeCap@224 | 14 | CLIP B16 | Patches | gaussian | 8.87 | 15.51 | 33.52 | 40.24 | 10.44 | 83.92 |
| DeCap@224 | 14 | CLIP B16 | Patches | attention | 4.81 | 11.64 | 25.96 | 24.47 | 7.34 | 80.67 |
| DeCap@592 | 37 | CLIP B16 | CLS | - | 22.43 | 24.64 | 49.25 | 84.57 | 18.66 | 90.36 |
| DeCap@592 | 37 | CLIP B16 | Patches | uniform | 9.75 | 15.79 | 34.40 | 42.06 | 10.63 | 84.04 |
| DeCap@592 | 37 | CLIP B16 | Patches | gaussian | 9.56 | 15.57 | 34.06 | 40.68 | 10.30 | 83.73 |
| DeCap@592 | 37 | CLIP B16 | Patches | attention | 5.50 | 11.80 | 26.78 | 24.46 | 7.21 | 80.65 |
| DeCap@224 | 7 | CLIP B32 | CLS | - | 23.46 | 25.12 | 50.06 | 87.40 | 19.14 | 90.58 |
| Patch-ioner@518 | 37 | DINOv2 B14 | CLS | - | 22.79 | 23.09 | 47.16 | 83.50 | 17.50 | 89.31 |
| Patch-ioner@518 | 37 | DINOv2 B14 | Patches | central patch | 15.68 | 18.46 | 40.84 | 55.53 | 12.66 | 84.26 |
| Patch-ioner@518 | 37 | DINOv2 B14 | Patches | uniform | 19.52 | 21.49 | 44.88 | 69.19 | 15.59 | 87.36 |
| Patch-ioner@518 | 37 | DINOv2 B14 | Patches | gaussian | 21.17 | 22.62 | 46.62 | 76.79 | 16.73 | 88.36 |
| Patch-ioner@518 | 37 | DINOv2 B14 | Patches | attention | 23.64 | 23.93 | 48.54 | 88.46 | 18.21 | 90.21 |
| Patch-ioner@518 GT Memory | 37 | DINOv2 B14 | CLS | - | 23.58 | 23.54 | 47.71 | 85.67 | 17.86 | 89.53 |
| Patch-ioner@518 GT Memory | 37 | DINOv2 B14 | Patches | attention | 25.66 | 24.77 | 49.83 | 93.87 | 19.09 | 90.70 |

Table 9. Image Captioning results on COCO test set.

(a) Trace Captioning (COCO)

| | Input | Weighting | B | M | R | C | S | P |
|---|---|---|---|---|---|---|---|---|
| $\mathcal{N}oise$ | CLS | - | 2.1 | 9.4 | 21.5 | 20.9 | 8.1 | 74.0 |
| $\mathcal{M}emory$ | CLS | - | 1.8 | 8.7 | 20.6 | 20.5 | 8.7 | 75.0 |
| $\mathcal{N}oise$ | Patches | uniform | **3.0** | **11.5** | **24.7** | **29.3** | 12.3 | 78.1 |
| $\mathcal{M}emory$ | Patches | uniform | 2.5 | 10.7 | 23.2 | 27.9 | **12.6** | **78.7** |

(b) Dense Captioning (Visual Genome v1.2)

| | Input | Weighting | mAP | M | B | R | C | S | P |
|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{N}oise$ | Patches | uniform | 0.20 | 10.36 | 1.24 | 17.76 | 26.34 | 12.63 | 76.98 |
| $\mathcal{M}emory$ | Patches | uniform | 0.21 | 10.63 | 1.36 | 18.59 | 31.94 | 15.03 | 78.82 |
| $\mathcal{N}oise$ | Patches | gaussian | 0.21 | 10.50 | 1.27 | 17.93 | 26.85 | 12.91 | 77.11 |
| $\mathcal{M}emory$ | Patches | gaussian | **0.22** | **10.82** | **1.43** | **18.82** | **32.80** | **15.48** | **79.14** |

(c) Region-Set Captioning (COCO Entities)

| | Input | Weighting | B | M | R | C | S | P |
|---|---|---|---|---|---|---|---|---|
| $\mathcal{N}oise$ | Patches | uniform | 10.5 | 18.4 | 37.2 | 97.5 | 26.7 | 85.6 |
| $\mathcal{M}emory$ | Patches | uniform | 11.5 | 19.3 | 38.8 | 109.1 | 29.4 | 87.5 |
| $\mathcal{N}oise$ | Patches | gaussian | 10.6 | 18.5 | 37.3 | 98.1 | 27.1 | 85.7 |
| $\mathcal{M}emory$ | Patches | gaussian | **11.6** | **19.6** | **39.3** | **111.6** | **30.1** | **87.7** |

(d) Image Captioning (COCO)

| | Input | Weighting | B | M | R | C | S | P |
|---|---|---|---|---|---|---|---|---|
| $\mathcal{N}oise$ | CLS | - | 20.88 | 22.29 | 46.16 | 74.78 | 15.96 | 86.52 |
| $\mathcal{M}emory$ | CLS | - | **22.79** | **23.09** | **47.16** | **83.50** | **17.50** | **89.31** |

(e) Image Captioning (Flickr30k)

| | Input | Weighting | B | M | R | C | S | P |
|---|---|---|---|---|---|---|---|---|
| $\mathcal{N}oise$ | CLS | - | 12.10 | 15.90 | 38.00 | 27.91 | 9.48 | 79.86 |
| $\mathcal{M}emory$ | CLS | - | **13.32** | **16.67** | **39.03** | **38.12** | **10.71** | **83.27** |

Table 10. **Mitigation to Modality Gap.** Memory-based Projection ($\mathcal{M}emory$) vs Noise-trained Decoder ($\mathcal{N}oise$).

which, unlike the original CLIP features, capture local semantic information. This is achieved by passing the image through a DINO model and utilizing the self-attention map of the CLS token to create proxy attention for the final CLIP visual layer.

In contrast, Talk2DINO directly maps CLIP embeddings to DINOv2 space, thus offering another approach to linking DINO's locally-rich representations with textual information. We tested ProxyCLIP with both DINO B/8 and DINOv2 B/14 backbones, while Talk2DINO was evaluated using the DINOv2 B/14 backbone. We trained the textual decoder on the CLIP B/16 embeddings for ProxyCLIP and on the Talk2DINO embeddings for the latter.

Results in Table 11 show that for zero-shot captioning tasks Talk2DINO achieves better performance.

## 9. Trace Captioning Benchmark Generation

As detailed in §4.1, we construct our Trace Captioning dataset from the Localized Narratives dataset [38]. This dataset consists of mouse traces and their corresponding transcriptions, where annotators describe objects in images while moving the mouse pointer over them.

The initial dataset samples include timestamped mouse traces and are composed of multiple sentences that thoroughly describe the trace, with the generated descriptions following the order of the mouse movement. However, our task does not require strict temporal coherence. Instead, we aim to generate a single, concise caption that describes the specific area covered by the localized trace, rather than a multi-sentence description.

To achieve this, we split the descriptions into individual sentences and align the traces accordingly. We then re-

(a) Trace Captioning (COCO)

| | Backbone | Weighting | B | M | R | C | S | P |
|---|---|---|---|---|---|---|---|---|
| ProxyCLIP | DINOv2 B/14 | uniform | 1.3 | 9.8 | 22.0 | 16.5 | 8.6 | 75.7 |
| ProxyCLIP | DINO B/8 | uniform | 1.3 | 9.8 | 22.1 | 16.7 | 8.5 | 75.7 |
| Talk2DINO | DINOv2 B/14 | uniform | **2.5** | **10.7** | **23.2** | **27.9** | **12.6** | **78.7** |

(b) Dense Captioning (Visual Genome v1.2)

| | Backbone | Weighting | mAP | M | B | R | C | S | P |
|---|---|---|---|---|---|---|---|---|---|
| ProxyCLIP | DINOv2 B/14 | uniform | 0.21 | 9.05 | 0.60 | 17.16 | 15.54 | 8.88 | 75.98 |
| ProxyCLIP | DINO B/8 | uniform | 0.21 | 9.09 | 0.59 | 17.20 | 15.69 | 8.88 | 76.00 |
| Talk2DINO | DINOv2 B/14 | uniform | 0.21 | 10.63 | 1.36 | 18.59 | 31.94 | 15.03 | 78.82 |
| ProxyCLIP | DINOv2 B/14 | gaussian | 0.21 | 9.21 | 0.62 | 17.34 | 16.11 | 9.18 | 76.17 |
| ProxyCLIP | DINO B/8 | gaussian | 0.21 | 9.28 | 0.63 | 17.44 | 16.36 | 9.24 | 76.23 |
| Talk2DINO | DINOv2 B/14 | gaussian | **0.22** | **10.82** | **1.43** | **18.82** | **32.80** | **15.48** | **79.14** |

(c) Region-Set Captioning (COCO Entities)

| | Backbone | Weighting | B | M | R | C | S | P |
|---|---|---|---|---|---|---|---|---|
| ProxyCLIP | DINO B/8 | uniform | 3.3 | 12.7 | 28.9 | 41.2 | 15.5 | 78.4 |
| ProxyCLIP | DINOv2 B/14 | uniform | 3.3 | 12.7 | 28.7 | 40.6 | 15.4 | 78.5 |
| Talk2DINO | DINOv2 B/14 | uniform | 11.5 | 19.3 | 38.8 | 109.1 | 29.4 | 87.5 |
| ProxyCLIP | DINO B/8 | gaussian | 3.4 | 13.0 | 29.2 | 42.8 | 16.1 | 78.8 |
| ProxyCLIP | DINOv2 B/14 | gaussian | 3.4 | 12.9 | 29.0 | 42.0 | 15.8 | 78.8 |
| Talk2DINO | DINOv2 B/14 | gaussian | **11.6** | **19.6** | **39.3** | **111.6** | **30.1** | **87.7** |

(d) Image Captioning (COCO)

| | Backbone | Weighting | B | M | R | C | S | P |
|---|---|---|---|---|---|---|---|---|
| ProxyCLIP | DINOv2 B/14 | uniform | 6.86 | 15.44 | 36.32 | 27.44 | 7.68 | 78.58 |
| ProxyCLIP | DINO B/8 | uniform | 6.99 | 15.59 | 36.64 | 28.70 | 7.81 | 78.99 |
| Talk2DINO | DINOv2 B/14 | uniform | 19.52 | 21.49 | 44.88 | 69.19 | 15.59 | 87.36 |
| ProxyCLIP | DINO B/8 | central patch | 7.83 | 15.84 | 37.38 | 34.19 | 7.99 | 79.21 |
| ProxyCLIP | DINOv2 B/14 | central patch | 8.42 | 16.19 | 37.72 | 36.30 | 8.39 | 79.65 |
| Talk2DINO | DINOv2 B/14 | central patch | 15.68 | 18.46 | 40.84 | 55.53 | 12.66 | 84.26 |
| ProxyCLIP | DINOv2 B/14 | gaussian | 7.30 | 15.85 | 36.97 | 30.69 | 8.05 | 79.35 |
| ProxyCLIP | DINO B/8 | gaussian | 7.32 | 15.87 | 37.07 | 30.99 | 8.10 | 79.58 |
| Talk2DINO | DINOv2 B/14 | gaussian | **21.17** | **22.62** | **46.62** | **76.79** | **16.73** | **88.36** |

Table 11. **Vision-Language Backbones.** Talk2DINO vs ProxyCLIP.

fine the traces by removing intermediate periods caused by transitions between sentences, which often occur when the annotator moves to a different region of the image. Specifically, we trim each trace by removing the first and last 15% of points, eliminating these transitional segments.

Furthermore, we refine the captions by prompting the Llama3 8B model to rephrase the sentences, removing vague or subjective phrases such as ”there is,” ”we can see,” or ”on the left of the image,” and replacing them with concise, objective descriptions that refer specifically to the region covered by the trace. This rephrasing is crucial to ensure that each caption adheres to the standard format of image-captioning datasets and focuses only on the precise part of the image that the trace corresponds to. The LLM also helps identify and remove irrelevant sentences (e.g., ”the image is blurred,” ”the image is edited”), which are then discarded along with their associated traces from the final benchmark.

For example, Figure 4 shows the full prompt used to guide the Llama model in refining and cleaning the descriptions. Figure 5 illustrates how the initial narrative samples are transformed into final trace captioning samples through the process of trace splitting and caption rephrasing.

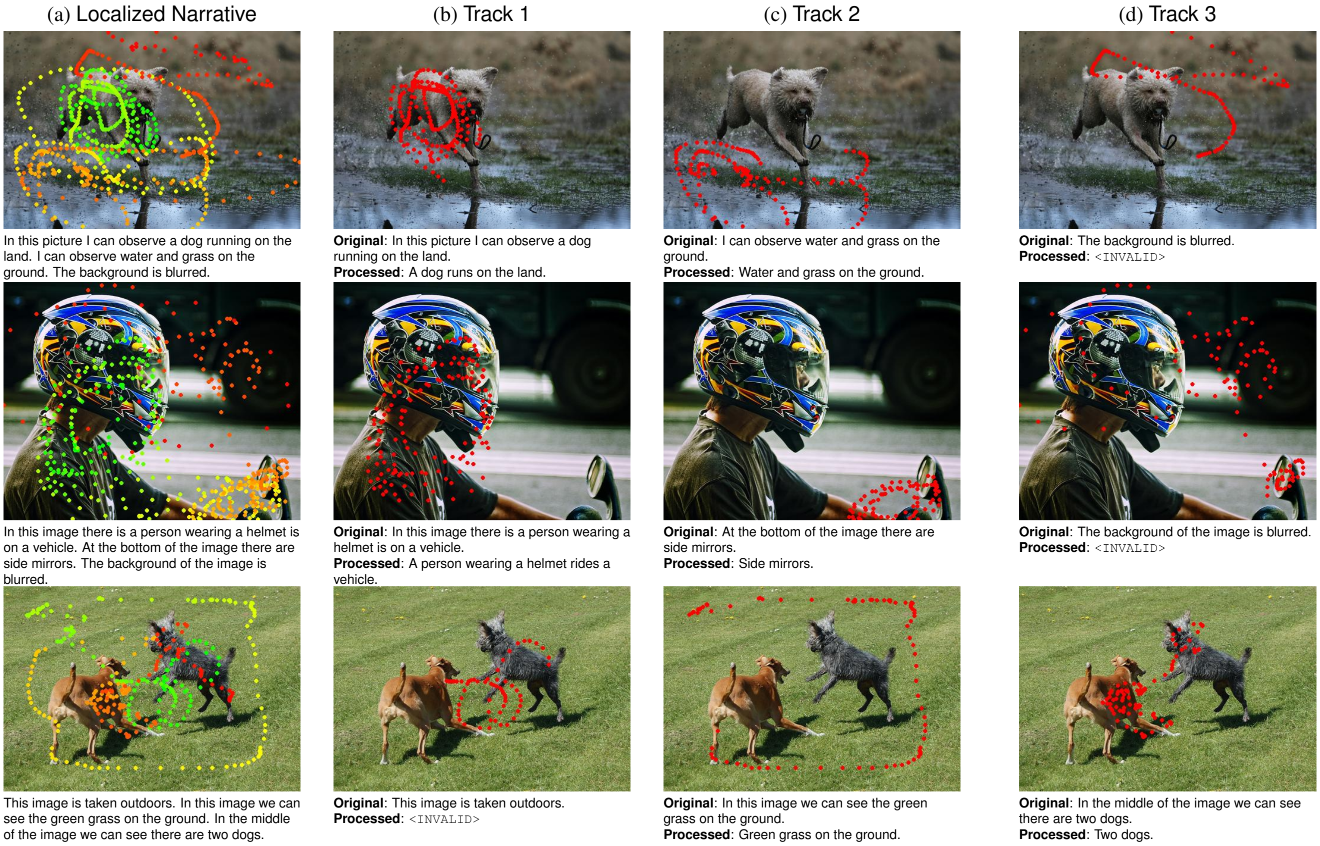
## 10. More Qualitative Results

Additional qualitative results are shown in Figure 6. Note that the first rows of Figures 3 and 6 contain also qualitative results for single patch captioning, for which we do not have annotated data to report quantitative results.

As can be noticed in Figures 3 and 6, the Region-Set Captioning task tends to align more closely with image-level captioning rather than strictly focusing on localized regions. This is expected since the ground-truth captions in the COCO Entities dataset originate from the image-level annotations of COCO, as stated in [7].

```
I have image descriptions derived from spoken narratives. These need to be rewritten as
concise, stand-alone captions in the style of the image-caption datasets. Follow these rules:

- Remove unnecessary narrative phrases like "we can see," "there is," "in this image," etc.
- Ensure the caption is standalone and descriptive.
- Use simple, objective language that highlights key elements.
- Keep it concise|just a single phrase.
- Follow the classical style of caption datasets.
- If the description is vague, subjective, or does not describe a concrete visual element
(e.g., "The image is taken indoor," "This image is blurred"), return ‘<INVALID>‘.
- Wrap the output in ‘{}‘ and add nothing else.

### **Examples:**
- **Input:** "We can see a young elephant stands which is near the water in a wooded area."
  **Output:** {A young elephant stands near the water in a wooded area.}

- **Input:** "In this image I can see some young children kicking a soccer ball in a field."
  **Output:** {A group of young children kicking a soccer ball around a field.}

- **Input:** "In the left of the image, we see a pole that has two green street signs on it."
  **Output:** {A pole has two green street signs on it.}

- **Input:** "We can see two surfboards which are stuck in the sand along the seashore."
  **Output:** {Two surfboards stuck in the sand along the seashore.}

- **Input:** "This image consists of a man which rides a wakeboard behind a boat."
  **Output:** {A man rides a wakeboard behind a boat.}

- **Input:** "In the background, there are a bunch of sticky notes and a pair of scissors."
  **Output:** {A bunch of sticky notes and a pair of scissors.}

- **Input:** "It looks like a sepia-toned photograph of a motorcycle underneath the shadow of a
tree."
  **Output:** {A sepia-toned photograph of a motorcycle underneath the shadow of a tree.}

- **Input:** "There is a sky"
  **Output:** {A sky.}

- **Input:** "She is smiling."
  **Output:** {A smiling girl.}

- **Input:** "The image is taken indoor."
  **Output:** {<INVALID>}

- **Input:** "This image is edited."
  **Output:** {<INVALID>}

- **Input:** "The image is blurred."
  **Output:** {<INVALID>}

- **Input:** "I think he is about to jump."
  **Output:** {<INVALID>}

Now, rewrite the following captions accordingly. Wrap each in ‘{}‘ and add nothing else:
<INPUT CAPTION>
```

Figure 4. **LLM Prompt for rephrasing trace captions**.



In this picture I can observe a dog running on the land. I can observe water and grass on the ground. The background is blurred.

**Original**: In this picture I can observe a dog running on the land.
**Processed**: A dog runs on the land.

**Original**: I can observe water and grass on the ground.
**Processed**: Water and grass on the ground.

**Original**: The background is blurred.
**Processed**: `<INVALID>`

In this image there is a person wearing a helmet is on a vehicle. At the bottom of the image there are side mirrors. The background of the image is blurred.

**Original**: In this image there is a person wearing a helmet is on a vehicle.
**Processed**: A person wearing a helmet rides a vehicle.

**Original**: At the bottom of the image there are side mirrors.
**Processed**: Side mirrors.

**Original**: The background of the image is blurred.
**Processed**: `<INVALID>`

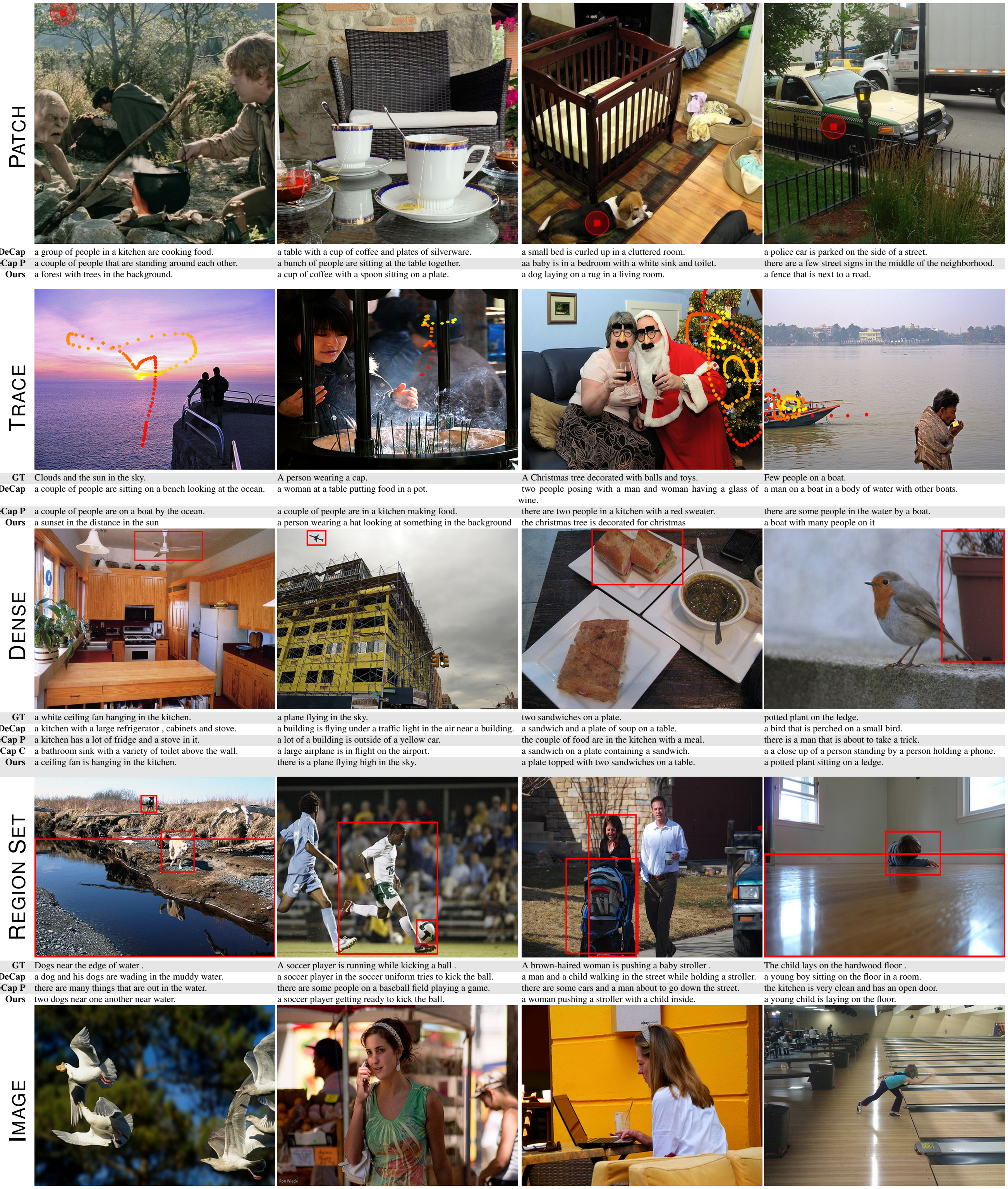
This image is taken outdoors. In this image we can see the green grass on the ground. In the middle of the image we can see there are two dogs.

**Original**: This image is taken outdoors.
**Processed**: `<INVALID>`

**Original**: In this image we can see the green grass on the ground.
**Processed**: Green grass on the ground.

**Original**: In the middle of the image we can see there are two dogs.
**Processed**: Two dogs.

Figure 5. **Narrative vs. Trace Samples**. The first column displays sample images from the localized narrative dataset. The remaining three columns show the corresponding mouse traces, along with the captions generated by the LLM. Captions marked with `<INVALID>` are removed from the dataset.



| | | | | |
|---|---|---|---|---|
| **DeCap** | a group of people in a kitchen are cooking food. | a table with a cup of coffee and plates of silverware. | a small bed is curled up in a cluttered room. | a police car is parked on the side of a street. |
| **DeCap P** | a couple of people that are standing around each other. | a bunch of people are sitting at the table together. | aa baby is in a bedroom with a white sink and toilet. | there are a few street signs in the middle of the neighborhood. |
| **Ours** | a forest with trees in the background. | a cup of coffee with a spoon sitting on a plate. | a dog laying on a rug in a living room. | a fence that is next to a road. |

| | | | | |
|---|---|---|---|---|
| **GT** | Clouds and the sun in the sky. | A person wearing a cap. | A Christmas tree decorated with balls and toys. | Few people on a boat. |
| **DeCap** | a couple of people are sitting on a bench looking at the ocean. | a woman at a table putting food in a pot. | two people posing with a man and woman having a glass of wine. | a man on a boat in a body of water with other boats. |
| **DeCap P** | a couple of people are on a boat by the ocean. | a couple of people are in a kitchen making food. | there are two people in a kitchen with a red sweater. | there are some people in the water by a boat. |
| **Ours** | a sunset in the distance in the sun | a person wearing a hat looking at something in the background | the christmas tree is decorated for christmas | a boat with many people on it |

| | | | | |
|---|---|---|---|---|
| **GT** | a white ceiling fan hanging in the kitchen. | a plane flying in the sky. | two sandwiches on a plate. | potted plant on the ledge. |
| **DeCap** | a kitchen with a large refrigerator , cabinets and stove. | a building is flying under a traffic light in the air near a building. | a sandwich and a plate of soup on a table. | a bird that is perched on a small bird. |
| **DeCap P** | a kitchen has a lot of fridge and a stove in it. | a lot of a building is outside of a yellow car. | the couple of food are in the kitchen with a meal. | there is a man that is about to take a trick. |
| **DeCap C** | a bathroom sink with a variety of toilet above the wall. | a large airplane is in flight on the airport. | a sandwich on a plate containing a sandwich. | a a close up of a person standing by a person holding a phone. |
| **Ours** | a ceiling fan is hanging in the kitchen. | there is a plane flying high in the sky. | a plate topped with two sandwiches on a table. | a potted plant sitting on a ledge. |

| | | | | |
|---|---|---|---|---|
| **GT** | Dogs near the edge of water . | A soccer player is running while kicking a ball . | A brown-haired woman is pushing a baby stroller . | The child lays on the hardwood floor . |
| **DeCap** | a dog and his dogs are wading in the muddy water. | a soccer player in the soccer uniform tries to kick the ball. | a man and a child walking in the street while holding a stroller. | a young boy sitting on the floor in a room. |
| **DeCap P** | there are many things that are out in the water. | there are some people on a baseball field playing a game. | there are some cars and a man about to go down the street. | the kitchen is very clean and has an open door. |
| **Ours** | two dogs near one another near water. | a soccer player getting ready to kick the ball. | a woman pushing a stroller with a child inside. | a young child is laying on the floor. |

| | | | | |
|---|---|---|---|---|
| **GT** | Four birds are chasing another bird which has a piece of food in its mouth. | Brown-haired girl wearing a green tank top, talking on a cellphone. | A woman with blond-hair is sitting in a booth with a drink working on her laptop. | A young girl in a blue shirt is in a bowling alley, and is casting her ball down a lane. |
| **DeCap** | a flock of birds flying over the water. | a woman talking on a cell phone while on a street. | a woman sitting at a table using a laptop. | a young girl playing a bowling game on wii. |
| **ZeroCap** | a gull mating. | a man in the back of a pickup truck with blood on the back. | a reader's writing on a laptop on desk-mounted computer. | a view hitting the deck pin at the end of the row stretch. |
| **CLOSE** | a group of birds flying over a body of water. | a woman looking at her cell phone while standing in a street. | a woman sitting at a table with a laptop and a drink. | on a person on a skateboard doing a trick. |
| **Ours** | a flock of birds flying in the sky. | a woman talking on a cell phone in a market. | a woman sitting at a cafe using her laptop. | a woman playing a bowling game on the bowling. |

Figure 6. **Qualitative results.** We report four predictions of our model and compare baselines from the finer (top) to the coarser (bottom) task. For trace captioning examples, the trace time is color-coded from start (red) to end (yellow). **DeCap** = DeCap applied on the whole image. **DeCap P** = DeCap applied on the same aggregation of patches used by our method. **DeCap C** = DeCap applied on cropped box. **ZeroCap** = ZeroCap [44] applied to the whole image. **CLOSE** = CLOSE [15] applied to the whole image. **GT** = ground-truth caption.