Explaining Caption-Image Interactions in CLIP models with Second-Order Attributions

Anonymous authors
Paper under double-blind review

Abstract

Dual encoder architectures like Clip models map two types of inputs into a shared embedding space and predict similarities between them. Despite their wide application, it is, however, not understood *how* these models compare their two inputs. Common first-order feature-attribution methods explain importances of individual features and can, thus, only provide limited insights into dual encoders, whose predictions depend on interactions between features.
In this paper, we first derive a second-order method enabling the attribution of predictions by any differentiable dual encoder onto feature-interactions between its inputs. Second, we apply our method to Clip models and show that they learn fine-grained correspondences between parts of captions and regions in images. They match objects across input modes and also account for mismatches. This intrinsic visual-linguistic grounding ability, however, varies heavily between object classes, exhibits pronounced out-of-domain effects and we can identify individual errors as well as systematic failure categories.

1 Introduction

Dual encoder models use independent modules to represent two types of inputs in a common embedding space and are optimized to predict a scalar similarity measure for them. The training objective is typically a triplet or contrastive loss (Sohn, 2016; van den Oord et al., 2019). Popular examples include Siamese transformers for text-text pairs (SBERT) (Reimers & Gurevych, 2019) and Contrastive Language-Image Pre-Training (Clip) models (Radford et al., 2021; Jia et al., 2021) for text-image pairs. The learned representations have proven to be highly informative for downstream applications, such as image classification (Zhang et al., 2022a), visual question answering (Antol et al., 2015; Tilli & Vu, 2025), image captioning and visual entailment (Shen et al., 2021), as well as text or image generation (Chen et al., 2023a; Yu et al., 2022; Rombach et al., 2022). In (multi-modal) information retrieval, dual encoders can be applied to perform efficient semantic search (Baldrati et al., 2022; Zhu et al., 2024). Their independent processing of the two inputs allows for the pre-computation and storage of representations in vector-databases enabling sub-linear search times via approximate nearest neighbor algorithms (Xiong et al., 2021; Johnson et al., 2019), which can serve Retrieval-Augmented Generation (RAG) Gao et al. (2023).

Despite the wiede-spread application of dual encoder models, an open question remains *how* these models compare the features of their two inputs. Ostensibly, common attribution methods can provide insights into such feature importances. However, different from single-input models, dual encoder predictions depend on feature interactions between two inputs, rather than on individual features. This is due to the final comparison of the two inputs' embeddings through a cosine-similarity or dot-product, resulting in all terms contributing to the output similairty score to contain multiplicative dependencies between the two inputs. In such multiplicative terms, a change in one involved feature affects the contribution of others. Therefore, we speak of *feature interaction*.
First-order feature attribution methods, like Shapley values (Lundberg & Lee, 2017) or integrated gradients (Sundararajan et al., 2017) cannot account for feature interactions because they can only attribute predictions to individual features (Zheng et al., 2020; Ramamurthy et al., 2022; Janizek et al., 2021; Sundararajan et al., 2020). Only few works have studied interactions of features in symmetric Siamese encoders (Eberle

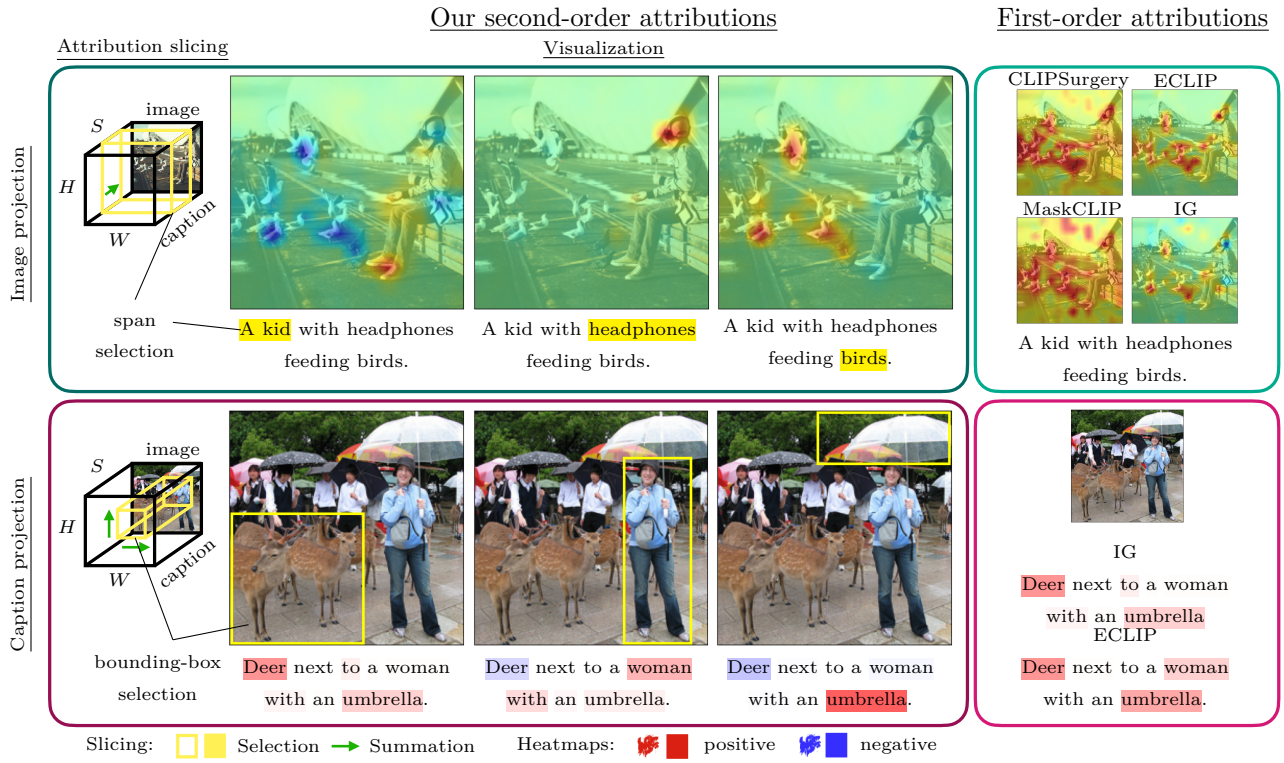


Figure 1: (Left column) Our second-order attributions can point out *interactions* between arbitrary spans in captions and regions in images. We can visualize them by slicing (yellow selection) our 3d attribution tensor with image dimensions (H, W) and caption dimension S (details in Section 3). A selection can be projected onto the image (top-left) or the caption (bottom-left) by summation (green arrows). Heatmaps for these projected attributions are in shades of red/blue for positive/negative values. (Right column) In contrast, first-order attributions can only attribute the overall similarity between captions and images onto either the image (top-right) *or* the caption (bottom-right). They *cannot* assess underlying interactions. Figure 21 provides an alternative illustration of the 3d aspect of our interaction attributions.

et al., 2020; Möller et al., 2023; 2024; Vasileiou & Eberle, 2024) and, to the best of our knowledge, they are yet to be explored in non-symmetric models like vision-language dual encoders such as Clip.

In this work, we address this research gap and aim at a means to analyze which aspects in two given inputs dual encoders compare to predict a similarity for them. Our contributions are the following:

(1) Motivated by the theory behind integrated gradients Sundararajan et al. (2017), we derive a general second-order feature attribution method that can explain *interactions* between inputs of any differentiable dual encoder model. The method does not rely on any modification of the trained model, nor on additional optimization. Required changes to the original code are minimal and easily transferable to different architectures (implementation details in Appendix H). We will make our code available.

(2) We apply the method to a range of Clip models and demonstrate that they can capture fine-grained interactions between corresponding parts of captions and regions in images. They identify matching objects across the input modes and also penalize mismatches. Using image-captioning datasets with object bounding-box annotations, we evaluate the extent and limitations of this *intrinsic visual-linguistic grounding ability* in a wide range of Clip models. We find large variation for different object classes and pronounced out-of-domain effects. An error analysis reveals typical failure categories.

Figure 1 illustrates our interaction attributions and puts them in relation to first-order variants.

2 Related work

Metric learning refers to the task of producing embeddings reflecting the similarity between inputs (Kaya & Bilge, 2019). Applications include face identification (Guillaumin et al., 2009; Wojke & Bewley, 2018) and image retrieval (Zhai & Wu, 2018; Gao et al., 2014). Siamese networks with cosine similarity of embeddings were early candidates (Chen & He, 2021). The triplet-loss (Hoffer & Ailon, 2015) involving negative examples has been proposed as an improvement but requires sampling strategies for the large number of possible triplets (Roth et al., 2020). Qian et al. (2019) have shown that the triplet-loss can be relaxed to a softmax variant. Sohn (2016) and van den Oord et al. (2019) have proposed the batch contrastive objective which has been applied in both unsupervised (Caron et al., 2020) and supervised representation learning (Khosla et al., 2020). It has lead to highly generalizable semantic text (Reimers & Gurevych, 2019) and image embeddings (He et al., 2020) and ultimately to the Clip training paradigm Radford et al. (2021).

Vision-language models process both visual and linguistic inputs. Zhang et al. (2022b) were the first to train a dual-encoder architecture with a contrastive objective on image-text data in the medical domain. With Clip Radford et al. (2021) have applied this principle to web-scale image captions and the ALIGN model has achieved similar results with alt-text (Jia et al., 2021). In the following, the basic inter-modal contrastive loss has been extended by, intra-modal loss terms (Goel et al., 2022; Lee et al., 2022; Yang et al., 2022a), self-supervision (Mu et al., 2022), non-contrastive objectives (Zhou et al., 2023), incorporating classification labels (Yang et al., 2022b), textual augmentation (Fan et al., 2023), a unified multi-modal encoder architecture (Mustafa et al., 2022) and retrieval augmentation (Xie et al., 2023). Next to more advanced training objectives, other works have identified the training data distribution to be crucial for performance: Gadre et al. (2023) have proposed the DataComp benchmark focusing on dataset curration while fixing model architecture and training procedure, Xu et al. (2024) have balanced metadata distributions and Fang et al. (2024) have introduced data filtering networks for the purpose. The strictly separated dual-encoder architecture has been extended to include cross-encoder dependencies (Li et al., 2022a; Pramanick et al., 2023), and multi-modal encoders have been combined with generative decoders (Chen et al., 2023a; Lu et al., 2023; Li et al., 2021; Koh et al., 2023; Alayrac et al., 2022). The CoCa model combines contrastive learning on uni-modal vision- and text-representations with a text generative cross-modal decoder (Yu et al., 2022).

Local feature attribution methods aim at explaining a given prediction by assigning contributions to individual input features (Murdoch et al., 2019; Doshi-Velez & Kim, 2017; Lipton, 2018; Atanasova et al., 2020). First-order gradients can approximate a prediction's sensitivity to such features (Li et al., 2016) and gradient$\times$input saliencies can approximate feature importances (Simonyan et al., 2014). In transformer architectures, attention weights were proposed as an explanation for model behavior (Abnar & Zuidema, 2020); however, subsequent works have contested this view, arguing that attention weights represent only one aspect of the model's reasoning (Jain & Wallace, 2019; Wiegreffe & Pinter, 2019; Bastings & Filippova, 2020). Layer-wise relevance propagation (LRP) defines layer-specific rules to back-propagate attributions to individual features (Montavon et al., 2019; Bach et al., 2015). In contrast, shapley values (Lundberg & Lee, 2017) and Integrated Gradients (IG) (Sundararajan et al., 2017) treat models holistically and can provide a form of theoretical guaranty for correctness. This has recently been challenged by Bilodeau et al. (2024) who prove fundamental limitations of attribution methods. A widely used attribution method in the vision domain is Gradient-weighted Class Activation Mapping (Grad-CAM) (Selvaraju et al., 2017), which Chefer et al. (2021) and Bousselham et al. (2024) extend to transformer architectures.
Assigning importances to individual features, first-order attribution methods cannot capture dependencies on feature interactions. Tsang et al. (2018) have proposed to detect such interactions from weight matrices in feed-forward neural networks, Cui et al. (2020) investigated them in Bayesian networks. The Shapley value has been extended to the Shapley (Taylor) Interaction Index (Grabisch & Roubens, 1999; Sundararajan et al., 2020; Fumagalli et al., 2024) and Janizek et al. (2021) have generalized IG to integrated Hessians. Dual and Siamese encoders represent a special case, as their predictions *only* depend on feature interactions due to their multiplicative cosine or dot-product comparison of the two inputs' embeddings (cf. Equation 1 below). Plummer et al. (2020) and Zheng et al. (2020) have assessed similarities in Siamese image encoders. Eberle et al. (2020) extended LRP for this class of models (Vasileiou & Eberle, 2024), while Möller et al.

(2023; 2024) extended IG to Siamese language encoders. In this work, we further generalize this method to multi-modal dual encoders.

CLIP explainability. Several works have previously pursued the goal of better understanding CLIP models and contrastive image encoders. Wang et al. (2023) and Kazmierczak et al. (2024) have proposed information bottleneck approaches. Bhalla et al. (2025) identified interpretable sparse concepts in the embedding space. Rasekh et al. (2024) predict human-understandable rationales for images. Quantmeyer et al. (2024) localized where the text encoder processes negation. Giulivi & Boracchi (2024) create saliency maps for WordNet concepts. Chen et al. (2022) propose an improved CAM variant and analyze which objects the model looks at. Materzyńska et al. (2022) are interested in the entaglement of image representations. Gandelsman et al. (2023) identified the roles of individual attention heads in CLIP's image encoder and later investivated second-order effects of neurons (Gandelsman et al., 2025). Lewis et al. (2024) analyzed whether CLIP models adequately handle compositional concepts. Sam et al. (2024) investigate the model's reasoning ability about differences in images. Tu et al. (2024) examined safety objectives in CLIP models and Mayilvahanan et al. (2024) investigated their out-of-domain generalization. Zhao et al. (2024) explored a wide range of first-order methods to attribute similarity scores to images and captions independently and Li et al. (2023) proposed the CLIPSurgery method. Sammani et al. (2023) and Lerman et al. (2021) independently introduced a second-order variant of gradCAM that can assess feature interactions. It can be applied to CLIP; in Appendix F, we show that it is a special case of our method. Most closely related to our work, Interaction Local Interpretable Model-Agnostic Explanations (InteractionLIME) (Joukovsky et al., 2023) pioneered the attribution of interactions between captions and images in CLIP models. However, relying on a local bilinear approximation of CLIP, it does not explain the original model and requires additional optimization and hyper-parameter tuning (cf. Appendix G). Last, the Image-Text Similarity Map (ITSM) by Li et al. (2022c) and the method by Black et al. (2022) are forward-facing saliency methods that compute importance values through pair-wise embedding multiplication. We compare these approaches against ours in Section 4.1.

Visual-linguistic grounding is the identification of fine-grained relations between text phrases and corresponding image parts (Chen et al., 2023b). Specialized models predict regions over images for a corresponding input phrase (Sadhu et al., 2019; Ye et al., 2019). This objective has been combined with contrastive caption matching (Li et al., 2022b; Datta et al., 2019), and caption generation (Yang et al., 2022c). The Vision-Language Transformer with weakly-supervised local-feature Alignment (VoLTA) model internally matches latent image-region and text-span representations (Pramanick et al., 2023). In multi-modal text generative models, grounding has been included as an additional pretraining task (Li et al., 2020; Su et al., 2019; Chen et al., 2020); alternatively grounding abilities can be unlocked with visual prompt learning (Dorkenwald et al., 2024). At the intersect of grounding and explainability, Hendricks et al. (2016) have generated textual explanations for vision models and have grounded them to input images (Hendricks et al., 2018; Park et al., 2018). In this paper, we do not optimize models to explicitly ground predictions, but aim at analyzing to which extent purely contrastively trained dual encoders already acquire this ability intrinsically.

3 Method

In the following, we derive general second-order attributions for dual encoder predictions enabling the assessment of feature-interactions between their two inputs.

Derivation of interaction attributions. Let

$$s = f(\mathbf{a}, \mathbf{b}) = \mathbf{g}(\mathbf{a})^\top \mathbf{h}(\mathbf{b}) \tag{1}$$

be a differentiable dual-encoder model, with two vector-valued encoders $\mathbf{g}$ and $\mathbf{h}$, respective inputs $\mathbf{a}$ and $\mathbf{b}$ and a scalar output s. For our purpose, $\mathbf{g}$ will be an image encoder with an image input $\mathbf{a}$ and $\mathbf{h}$ will be a text encoder with a text representation $\mathbf{b}$ as input. To attribute the prediction s onto features of the two inputs $\mathbf{a}$ and $\mathbf{b}$, we also define two uninformative *reference* inputs $\mathbf{r}_a$, the black image, and $\mathbf{r}_b$, a sequence of

fixed length padding tokens. We then rigorously start from the following expression:

$$f(\mathbf{a}, \mathbf{b}) - f(\mathbf{r}_a, \mathbf{b}) - f(\mathbf{a}, \mathbf{r}_b) + f(\mathbf{r}_a, \mathbf{r}_b) \tag{2}$$

Our derivation first proceeds by showing the equality of this initial starting-point to Eq. 10. We then reduce this equality to our final attributions in Eq. 11 using the approximations discussed below. At this point, we are also discussing an intuitive interpretation of the final result. As a first step, however, we proceed rigorously, see f as an anti-derivative and can turn the above expression into an integral over the derivative of f:

$$\begin{aligned} &\big[f(\mathbf{a}, \mathbf{b}) - f(\mathbf{r}_a, \mathbf{b})\big] - \big[f(\mathbf{a}, \mathbf{r}_b) - f(\mathbf{r}_a, \mathbf{r}_b)\big] \\ &= \int_{\mathbf{r}_b}^{\mathbf{b}} \frac{\partial}{\partial \mathbf{y}_j} \big[f(\mathbf{a}, \mathbf{y}) - f(\mathbf{r}_a, \mathbf{y})\big] \, d\mathbf{y}_j = \int_{\mathbf{r}_b}^{\mathbf{b}} \int_{\mathbf{r}_a}^{\mathbf{a}} \frac{\partial^2}{\partial \mathbf{x}_i \partial \mathbf{y}_j} f(\mathbf{x}, \mathbf{y}) \, d\mathbf{x}_i \, d\mathbf{y}_j \end{aligned} \tag{3}$$

Here, $\mathbf{x}$ and $\mathbf{y}$ are integration variables for the two inputs. We use component-wise notation with indices i and j for the input dimensions and omit sums over double indices for clarity. We plug in the model definition from Equation 1:

$$\int_{\mathbf{r}_a}^{\mathbf{a}} \int_{\mathbf{r}_b}^{\mathbf{b}} \frac{\partial^2}{\partial \mathbf{x}_i \partial \mathbf{y}_j} \mathbf{g}_k(\mathbf{x}) \, \mathbf{h}_k(\mathbf{y}) \, d\mathbf{x}_i \, d\mathbf{y}_j \tag{4}$$

Again, we use component-wise notation for the dot-product between the two embeddings $\mathbf{g}(\mathbf{x})$ and $\mathbf{h}(\mathbf{y})$ and index output dimensions with k. Since neither embedding depends on the other integration variable, we can separate the integrals:

$$\int_{\mathbf{r}_a}^{\mathbf{a}} \frac{\partial \mathbf{g}_k(\mathbf{x})}{\partial \mathbf{x}_i} \, d\mathbf{x}_i \int_{\mathbf{r}_b}^{\mathbf{b}} \frac{\partial \mathbf{h}_k(\mathbf{y})}{\partial \mathbf{y}_j} \, d\mathbf{y}_j \tag{5}$$

This step makes explicit use of the strict independence of the two encoders. Cross-encoder architectures would introduce dependencies between them. Both terms are line integrals from the references to the actual inputs in the respective input representation spaces; $\partial \mathbf{g}_k(\mathbf{x}) / \partial \mathbf{x}_i$ and $\partial \mathbf{h}_k(\mathbf{y}) / \partial \mathbf{y}_j$ are the Jacobians of the two encoders. To proceed with these integrals, we define integration paths and solve by substitution. We follow Sundararajan et al. (2017), and use the straight lines between both references and inputs,

$$\mathbf{x}(\alpha) = \mathbf{r}_a + \alpha(\mathbf{a} - \mathbf{r}_a), \tag{6}$$

$$\mathbf{y}(\beta) = \mathbf{r}_b + \beta(\mathbf{b} - \mathbf{r}_b), \tag{7}$$

parameterized by α and β. For the integral over encoder $\mathbf{g}$ substituting the path $\mathbf{x}(\alpha)$ yields an integral over the the scalar integration variable α:

$$\int_0^1 \frac{\partial \mathbf{g}_k\left(\mathbf{x}(\alpha)\right)}{\partial \mathbf{x}_i} \frac{\partial \mathbf{x}_i(\alpha)}{\partial \alpha} \, d\alpha = (\mathbf{a} - \mathbf{r}_a)_i \int_0^1 \frac{\partial \mathbf{g}_k\left(\mathbf{x}(\alpha)\right)}{\partial \mathbf{x}_i} \, d\alpha \tag{8}$$

Since $\partial \mathbf{x}(\alpha) / \partial \alpha = (\mathbf{a} - \mathbf{r}_a)$ is a constant w.r.t α, we can pull it out of the integral. We then define the *integrated Jacobian*,

$$\mathbf{J}^a_{ki} = \int_0^1 \frac{\partial \mathbf{g}_k\left(\mathbf{x}(\alpha)\right)}{\partial \mathbf{x}_i} \, d\alpha \approx \frac{1}{N} \sum_{n=1}^{N} \frac{\partial \mathbf{g}_k(\mathbf{x}(\alpha_n))}{\partial \mathbf{x}_i}, \tag{9}$$

as the analogon to integrated gradients for vector-valued models. The integral over encoder $\mathbf{h}$ can be processed in the same way by substituting $\mathbf{y}(\beta)$ to obtain $\mathbf{J}^b_{kj}$. In practice, these integrals are calculated numerically by sums over N steps, with $\alpha_n = n/N$. This introduces an approximation error which must, however, converge to zero for large N by definition of the Riemann integral. We plug the results from Equation 8 and the definitions of the *integrated Jacobians* into Equation 5:

$$(\mathbf{a} - \mathbf{r}_a)_i \, \mathbf{J}^a_{ik} \mathbf{J}^b_{kj} \, (\mathbf{b} - \mathbf{r}_b)_j =: \mathbf{A}_{ij} \tag{10}$$

After computing the sum over the output embedding dimensions k, this provides a matrix of interactions between feature-pairs (i, j) in input $\mathbf{a}$ and $\mathbf{b}$, respectively, which we call the *attribution matrix* $\mathbf{A}_{ij}$. This

final result is the product between the integrated Jaobians of the two encoders, thus, reducing the initial second-order derivative from Eq. 3 to a product of two first-order derivatives.
Note that except for the numerical integration, the equality to Equation 2 still holds. Hence, the sum over all feature-interaction attributions in $\mathbf{A}$ is an exact reformulation of our starting-point. If the references $\mathbf{r}_a$ and $\mathbf{r}_b$ are uninformative, i.e. $f(\mathbf{r}_a, \mathbf{b}) \approx 0$, $f(\mathbf{a}, \mathbf{r}_b) \approx 0$, $f(\mathbf{r}_a, \mathbf{r}_b) \approx 0$, we can reduce the equality between Equations 2 and 10 to the following approximation:

$$f(\mathbf{a}, \mathbf{b}) \approx \sum_{ij} \mathbf{A}_{ij}. \tag{11}$$

Intuitively, this provides an approximate decomposition of the model prediction $s = f(\mathbf{a}, \mathbf{b})$ into additive contributions of feature-pair interactions between the two inputs.
It can be thought of as the natural extension of integrated gradients (Sundararajan et al., 2017) to dual encoder models that compute similarities between two inputs based on the distance of their embeddings. Throughout this paper, we evaluate the attribution matrix $\mathbf{A}$. Below, we describe which form it takes in CLIP models and how we proceed to visualize it.

Inter-modal attributions. In the derivation above, we treat image and text representations as vectors. In transformer-based encoders, text inputs are represented as $S \times D_b$ dimensional tensors, where S is the length of the token sequence. Image representations are of shape $H \times W \times D_a$, with H and W being height and width of the image representation; in vision-transformers both equal the number of patches P. D_a and D_b are the encoders' embedding dimensionalities. Our pair-wise image-text interaction attributions thus have the dimensions $H \times W \times D_b \times S \times D_a$, which quickly becomes intractably large. Fortunately, the sum over dimensions in Equation 11 enables the additive combination of attributions in $\mathbf{A}$. We sum over the embedding dimensions of both encoders D_a and D_b and obtain a $H \times W \times S$ dimensional attribution tensor, which estimates for *each pair of a text token and an image patch* how much their combination contributes to the overall prediction. These attributions are still three-dimensional and thus not straightforward to visualize. However, again we can use their additivity, slice the 3d attribution tensor along text or image dimensions and project onto the remaining dimensions by summation. This projection is demonstrated in Figure 1 schematically and with examples, both for a selection over a token range in the caption (top) and a selection over a bounding-box in the image (bottom). Importantly, all three examples per caption-image pair come from the same 3d attribution tensor.

Intra-modal attributions. Albeit vision-language dual-encoders are typically trained to match images against captions, we can compute attributions for image-image or text-text pairs as well by applying the same encoder to both inputs. For text-text attributions, after summation over embedding dimensions, this yields an $S_1 \times S_2$ dimensional attribution tensor, with S_1 and S_2 being token sequence lengths of the two texts. Analogous to above, in Figure 2 (left) we attribute the yellow selected slice in the top caption onto the bottom caption. For image-image similarities, attribution tensors become four dimensional taking the shape $(H \times W)_1 \times (H \times W)_2$ and containing a contribution for every pair of two patches from either image. In Figure 2 (right), we attribute the slice of the yellow bounding-box in the left image onto the image to its right. Appendix A includes additional examples.

4 Experiments

In our experiments, we apply our feature-interaction attributions to CLIP models. We focus on evaluating the interactions between mentioned objects in captions and corresponding regions in images by selecting token-ranges in captions and analyzing their interactions with image patches. In the first series of experiments, we compare our attributions against baselines (Section 4.1). The second series in Section 4.2 then utilizes our method and analyzes CLIP models.

Datasets. We base this evaluation on three image-caption datasets that additionally contain object bounding-box annotations in images, Microsoft's Common Objects in Context (COCO) (Lin et al., 2014), the Flickr30k collection (Young et al., 2014; Plummer et al., 2015), and the Hard Negative Captions (HNC)

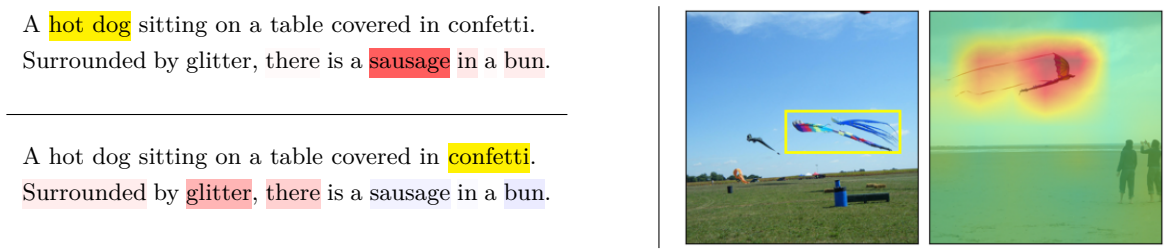


Figure 2: (Left) Intra-modal text-text attributions between top and bottom captions (top: selections in yellow, bottom: corresponding attributions in red/blue for positive/negative). (Right) Intra-modal image-image attributions between left and right image (left: bounding-box selection in yellow, right: heatmaps as above). More examples can be found in Figure 10.

dataset by Dönmez et al. (2023). For HNC, we apply the authors' approach of generating captions from scene graphs using templates. Specifically, we use a basic template of the form `subject predicate object` to align the generated captions with the domain of the other two datasets. We use HNC for evaluation only, on Flickr30k we use the test split, and on COCO we use the validation split as the test split does not contain captions[1].

Models. We analyze CLIP dual-encoders (Radford et al., 2021) trained with the standard inter-modal contrastive objective. We evaluate the original OPENAI models, as well as METACLIP (Xu et al., 2024) and the OPENCLIP reimplementations trained on the LAION(Schuhmann et al., 2022), DFN(Fang et al., 2024), COMMONPOOL, and DATACOMP(Gadre et al., 2023) datasets [2].

Fine-tuning. In addition to the unmodified models, we evaluate variants fine-tuned on the COCO and Flickr30k training splits. We run all tuning for five epochs using AdamW (Loshchilov & Hutter, 2018), starting with an initial learning rate of 1×10^{-7} that exponentially increases to 1×10^{-5}. We set the weight decay to 1×10^{-4} and use a batch size of 64 on a single 50GB NVIDIA A6000.

4.1 Attribution evaluation

In the first series of experiments, we compare our attributions against baselines. Figure 1 shows a qualitative comparison of our second-order feature-interaction attributions against first-order variants, demonstrating how our method can point out correspondence between image regions and spans in a caption, while the latter can only attribute the overall similarity onto parts of the image or the caption. A detailed comparison between first-order methods has been presented by Zhao et al. (2024). We closely follow their evaluation protocol and extend it to second-order methods. Unless stated otherwise, we attribute to the second-last hidden representation in the models' image and text encoders and use $N = 50$ integration steps with a black image as the image reference and a padding token sequence as the text reference. In Appendix C, we include additional experiments on the accuracy of our attributions as a function of N, as well as different reference choices.

Baselines. We compare our method against four baselines, namely Interaction Class Activation Mapping (Interaction-CAM), InteractionLIME, ITSM and a variant of it. Interaction-CAM (Sammani et al., 2023) is also gradient-based and can be seen as a special case of our approach as shown in Appendix F. InteractionLIME is a bilinear extension of LIME for dual-encoder models (Joukovsky et al., 2023). Code is not available, therefore, we reimplement it; details are in Appendix G. ITSM (Sammani et al., 2023) follows the simple approach of pair-wise multiplication of token and image patch embeddings after applying CLIP's final projection layer to the individual embeddings. Originally, it is applied to output representations and we refer to this variant as ITSM_{out}. We also apply it to the same hidden representations that our method attributes

[1]https://www.kaggle.com/datasets/shtvkumar/karpathy-splits
[2]CLIP family: https://github.com/openai/CLIP, Open family: https://github.com/mlfoundations/open_clip

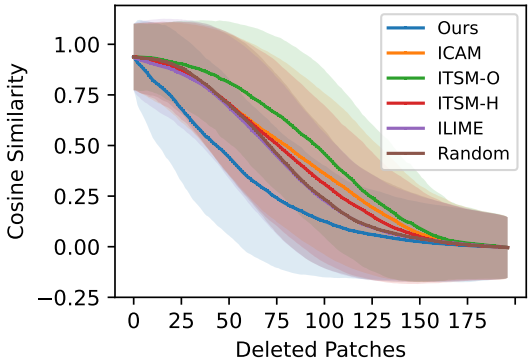


Figure 3: Decline of average similarity scores with standard deviations as uncertainty intervals between images and captions for iterative conditional image patch deletions using a model pretrained on LAION and fine-tuned on COCO. Additional figures displaying CID, CII, CTD, and CTI can be found in Fig. 16 and 17.

	Ours	ICAM	ITSM-O	ITSM-H	ILIME	Random
			LAION(tuned)			
CID ↓	**65.94**	91.39	101.35	88.88	83.43	85.04
CII ↑	**113.35**	80.21	69.01	87.53	86.48	85.18
CTD ↓	4.32	7.86	6.28	7.41	**4.05**	7.52
CTI ↑	9.26	7.77	7.82	6.72	**9.89**	7.47
			OPENAI			
CID ↓	**16.33**	21.04	23.97	20.91	20.25	20.47
CII ↑	**24.43**	20.47	16.24	20.60	20.58	20.47
CTD ↓	**1.06**	1.35	1.15	1.30	1.07	1.31
CTI ↑	**1.06**	0.95	0.99	0.92	1.04	0.96

Table 1: The AUC for CID, CII, CTD, and CTI, on COCO for the fine-tuned LAION and the original OPENAI model. ↓: lower is better; ↑: higher is better. Corresponding plots in Fig. 16.

to and refer to this variant as ITSM_{hidden}. A qualitative comparison between all methods is included in Figure 11.

Input perturbation. Following Sammani et al. (2023), we perform conditional perturbation experiments by iteratively removing or inserting the most attributed features in one input while keeping the other input unmodified. Figure 3 plots the decrease in similarity score for conditional image patch deletion (CID). Our method produces the steepest score decline as a function of the number of patches removed, indicating its ability to identify the most relevant interactions. Next to CID, we also evaluate conditional image patch insertion (CII) as well as conditional text token deletion (CTD) and conditional text token insertion (CTI). All plots are shown in Figures 16 and 17. Table 1 provides a summary and reports the area under the curve (AUC) for the four variants. With the exception of InteractionLIME on the text side, our method consistently results in the highest AUC values for the insertion experiments and the lowest for deletion. While InteractionLIME performs good on conditional text attribution, interestingly, its image attributions are not competitive. We discuss this in Appendix G. Insertion and deletion experiments have been criticized for producing out-of-domain inputs Hooker et al. (2019). Therefore, we also construct in-domain perturbations in the form of *hard negative captions* (HNC) and evaluate their effect on the model in Section 4.2.

Object localization. To systematically assess the visual-linguistic grounding abilities of the analyzed dual encoders, we evaluate the models' localization ability of objects in images that are also mentioned in a given caption. For this experiment, we include all object annotations that correspond to a single instance of its class in the image, and whose bounding-box is larger than one patch. For COCO, we identify class occurrences in the caption through a dictionary based synonym matching. For HNC, classes exactly match sub-strings in captions and in Flickr30k, respective spans are already annotated. This results in 3.5k image-caption pairs from COCO, 8k pairs from Flickr30k, and 500 pairs from HNC. We compute attributions between the token span to a class mention in the caption and the image. Following Zhao et al. (2024), we then employ the Point Game (PG) framework by Zhang et al. (2018) to evaluate how well attributions correspond to human bounding-box annotations. It defines Point Game Accuracy (PGA) as the fraction of cases where the most attributed patch falls within the objects' bounding-box, and Point Game Energy (PGE) as the fraction of positive attributions within the bounding-box relative to the total attribution across the entire image (Zhao & Chan, 2023; Wang et al., 2020). For the latter we compare both full distributions (Figure 4 (right)) and median values (mPGE). The PG-framework is particularly suitable to evaluate the agreement between attributions and bounding-box annotations because it does not require any thresholding and post-processing

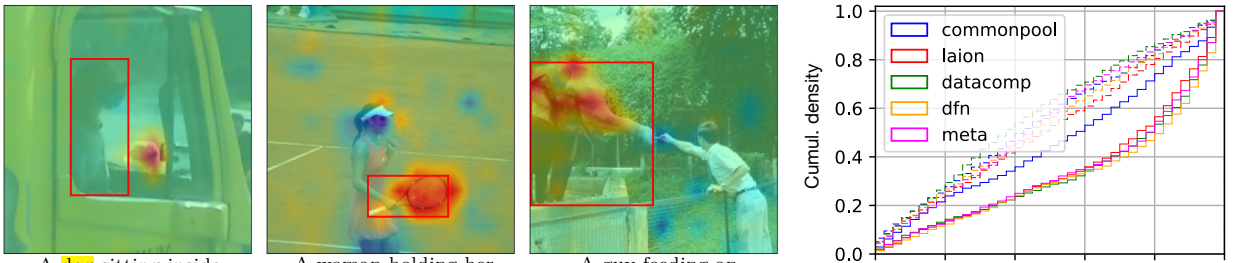


Figure 4: (Left) Examples for attributions between selected objects in the caption (yellow) and the image together with corresponding Coco bounding-boxes (red), PGE and PGA values as described in Section 4.1. (Right) Cumulative PGE distributions for the OpenClip models on Coco before (dashed) and after (solid) in-domain fine-tuning.

Training	Method	Coco		Flickr30k	
		mPGE	PGA	mPGE	PGA
OpenAI	ITSM_{out}	18.1	21.4	19.5	23.3
	ITSM_{hidden}	29.8	38.1	29.5	37.4
	ILIME	27.9	34.9	25.8	33.1
	ICAM	38.6	54.6	33.5	51.4
	Ours	**72.3**	**79.0**	**64.4**	**72.1**
Laion (tuned)	ITSM_{out}	22.8	30.3	24.5	28.7
	ITSM_{hidden}	30.5	34.6	28.8	36.6
	ILIME	28.8	37.8	25.8	34.5
	ICAM	32.5	58.4	33.5	51.4
	Ours	**71.1**	**83.2**	**54.3**	**61.8**

(a) PG-based comparison of our attributions against the ITSM method and Interaction-CAM (ICAM) for the OpenAI and Laion model.

Train	Tuning	Coco		Hnc		Flickr30k	
		mPGE	PGA	mPGE	PGA	mPGE	PGA
OpenAI	No	72.3	79.0	57.0	65.0	64.4	72.1
	Yes	**78.0**	**82.9**	-	-	**73.4**	**79.0**
Laion	No	49.4	63.3	40.0	51.6	38.2	52.0
	Yes	**71.1**	**83.2**	-	-	**54.6**	**61.8**

(b) Results for the Point Game-based vision-language grounding evaluation for the ViT-B-16 models trained by OpenAI and on Laion. *Tuning* indicates whether a model was fine-tuned on the train split of a dataset. Improvement upon fine-tuning are in bold.

Table 2: PGA: Point Game Accuracy, mPGE: median Point Game Energy. Extensive results for Table 2a including additional models are shown in Table 5. Full results of Table 2b can be found in Tables 3 and 4.

to construct bounding-box estimates that are then compared to annotations by means of intersection over union or similar metrics.

Figure 4 shows examples from different PGE-ranges and the corresponding cumulative distributions over the Coco dataset for the OpenClip models. Very high or low values, unambiguously indicate object correspondence or clear failure cases, respectively. However, intermediate values often result from attributions extending beyond the bounding box to contextual elements, such as the *tennis court* in the second example.

Table 2a compares our method against Interaction-CAM and ITSM for the OpenAI and Laion model. Results for DataComp and Dfn are included in Table 5. Our attributions outperform the baselines by large margins. Figure 15 includes cumulative PGE-distributions for our attributions and the baselines, evaluated on a selection of models using the Coco test split. Based on these distributions, we test whether the improvement of our method over the baselines is statistically significant using the framework for stochastic order proposed by Dror et al. (2019) (details in Appendix E). At the strict criterion of $p < 0.001$ and $\epsilon = 0.01$, our method clearly results in significantly better PGE-statistics. This shows that neither a simplified gradient-based approach (Interaction-CAM), nor pair-wise embedding multiplication (ITSM) or the optimization of a local surrogate model (InteractionLIME) can capture caption-image interactions in Clip models as well as our method.

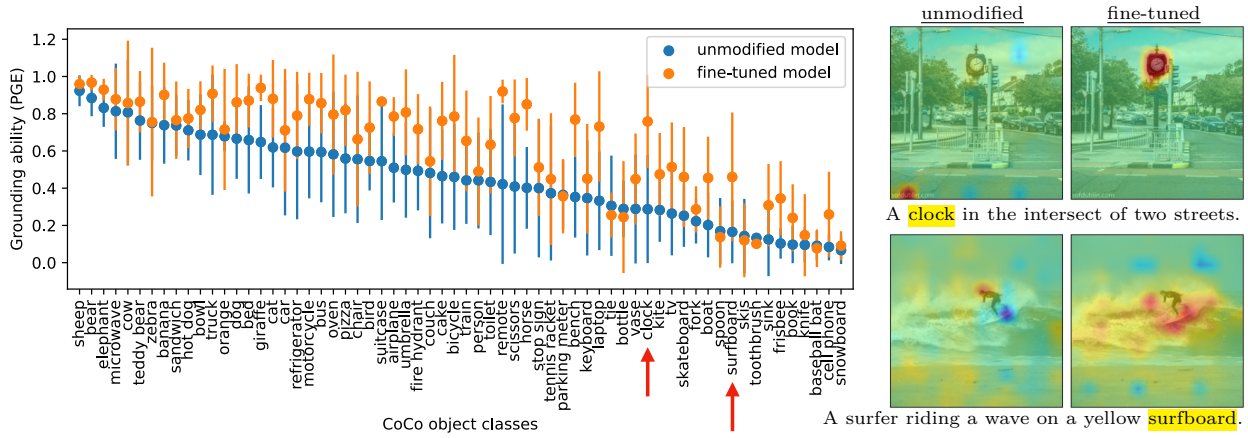


Figure 5: Class-wise average Point Game Energy (PGE) and its standard deviation (error bars) of the OpenClip Laion model before and after in-domain fine-tuning on the Coco train split. On the right are two explicit examples of how the model's grounding ability changes upon tuning. The corresponding classes are emphasized with red arrows.

4.2 Model analysis

We now turn to applying our method to gain insights into how Clip models match images and captions.

Out-of-domain effects. The tested models are trained on large web-based captioning datasets but have (presumably) not been tuned on the Flickr30k and Coco train splits. To assess domain effects of the models' grounding ability, we fine-tune them on the respective train splits. We emphasize that all fine-tuning is conducted within the standard contrastive framework, neither modifying model architectures nor training objectives to explicitly perform grounding. Table 2b presents the median PGE and PGA for the OpenAI model and its OpenClip Laion counterpart, before and after fine-tuning. Figure 4 shows cumulative PGE distributions before and after fine-tuning for the OpenClip models on the Coco dataset. The results for all tested OpenAI and OpenClip models on all datasets are provided in Appendix B. To compare the grounding ability of unmodified models and their fine-tuned counterparts, again we test whether one PGE-distribution is stochastically larger than the other (cf. Appendix E), assuming $p < 0.001$ and $\epsilon = 0.01$.

For both OpenAI and OpenClip models, fine-tuning increases grounding abilities by a large margin. These improvements are consistently significant. While the unmodified Clip ViT-B/16 model already demonstrates strong grounding abilities on Coco and Flickr30k, the off-the-shelf OpenClip counterparts perform notably worse on these datasets. However, their improvement after in-domain fine-tuning is remarkable, which is apparent in the examples in Figure 5. The off-the-shelf model fails to identify the *clock* and even assigns a negative attribution to the *surfboard*, whereas the fine-tuned version clearly identifies both. This large improvement in the models' grounding abilities indicates limitations in the original models' abilities to generalize beyond the initial training domain.

Class-wise evaluation. To examine the models' understanding of individual visual-linguistic concepts on a more fine-grained level and how it evolves upon in-domain tuning, we break the above analysis down to individual classes. Figure 5 shows the average PGE-values and their standard deviations for Coco classes in the OpenClip Laion model. The classes are ordered from left to right based on their average grounding ability in the unmodified model (blue). PGE values range from 0.92 ± 0.08 for *sheep* to 0.07 ± 0.07 for *snowboard*. The model effectively identifies the leftmost classes *sheep*, *bear*, *elephant*, while grounding is notably weaker for the rightmost classes like *snowboard*, *cell phone*, *baseball bat*. Upon fine-tuning (orange), most classes show improvements. Using the standardized mean difference of the two PGE values as a measure for effect size, we observe the largest improvements for the classes *horse*, *bench*, *giraffe*, *airplane* and *clock*. This shows that standard contrastive fine-tuning sharpens the fine-grained visual-linguistic correspondence

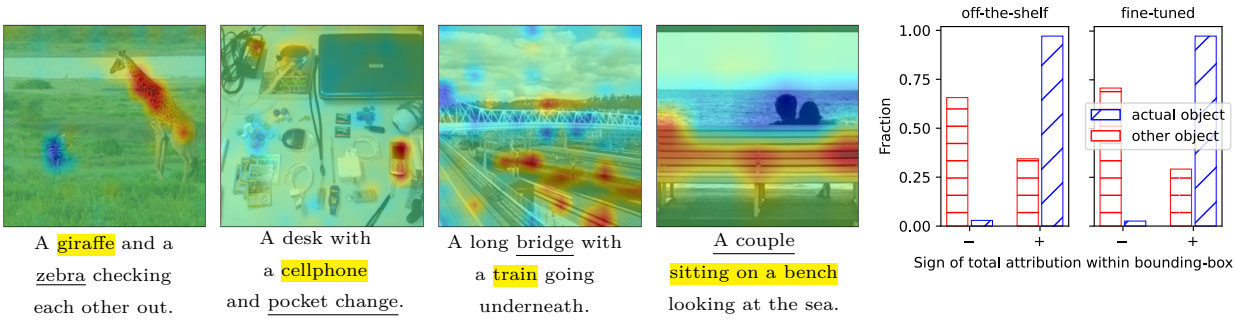


Figure 6: Attributions between selected parts of a caption (yellow) and a corresponding image. Other objects that also appear in the image and are mentioned in the caption (underlined) but are not selected for attribution often receive negative attributions (blue). The histogram on the right shows the distribution over the sign of such cross-attributions as described in the parageraph *Object Discrimination*.

of individual concepts in CLIP models. In Appendix B (Figure 18), we replicate this experiment for DFN and COMMONPOOL, yielding similar results.

Object Discrimination. We frequently observe that attributions between a given object in the text and a non-matching one in the image – or vice versa – are not only neutral but negative. Figure 6 includes four explicit examples. To systematically evaluate this effect, we sample instances from COCO that include at least two distinct object classes, each appearing exactly once in the image. We then compute attributions between the two corresponding bounding-boxes and text spans and also across them, which we refer to as cross-attribution. Attribution to the actual object's bounding-box is positive in nearly all cases (97.1%), while cross-attributions to the other object are negative in 65.6% of instances – rising to 70.1% in the COCO fine-tuned model (cf. Figure 6 (right)). This implies that the models do not only match corresponding objects across the input modes but can also actively penalize mismatches by assigning them negative contributions.

Hard negative captions. On the text side, it is straightforward to produce in-domain perturbations. We create hard negative captions that replace a single object in a positive caption with a reasonable but different object to receive a negative counterpart. To this end, we leverage the automatic procedure by (Dönmez et al., 2023) together with our simplified template (cf. Section 4) and additionally create a second resource from COCO by manually annotating a small yet high-quality evaluation sample of 100 image-caption pairs. We check whether our negative captions actually result in a decrease of the predicted similarity score compared with their positive counterparts and define the difference as δ_S. It is negative in 95.2% (89.1%) of the COCO (HNC) pairs. We then compute attributions between the token range of the original or replaced object and the object bounding-box in the image and define the attribution difference between the negative and the positive caption as δ_A. It is also negative in 95.2% (74.1%) of the COCO (HNC) examples. Full histograms for δ_S and δ_A as well as an example for a change in attributions is included in Figure 8. These results show that the model mostly reacts correctly to mistakes in the caption and decreases the attribution to the corresponding image region.

Qualitative failure analysis. To identify cases where the models' grounding abilities are systematically weak, we extract objects with PGE < 0.2 from the COCO validation set and categorize them qualitatively. For the LAION, OPENAI , and DFN models, this results in approximately 200 image-caption pairs each. We can identify five major categories: (1) Visually *correlated scenes* like baseball courts, bathrooms, offices, etc., (2) attributions *locally exceeding bounding boxes*, (3) *coverage or partial visibility* of objects, (4) actual *object misidentifications*, and (5) *difficult or unusual scenes*. Figure 7 shows the distribution among these categories and an example for each. More examples are included in Figure 12. Category (1), *correlated scenes*, accounts for approximately half of all failures in all three models, indicating that one of weaknesses of CLIP models is to differentiate between objects that frequently appear together.

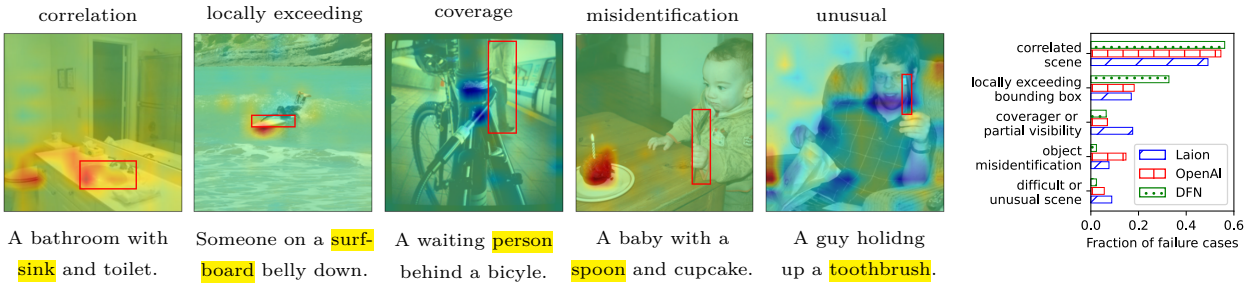


Figure 7: Examples for the five failure categories that we can identify (left) and their relative occurrence in three models (right). More examples for all categories are in included in Figure 12.

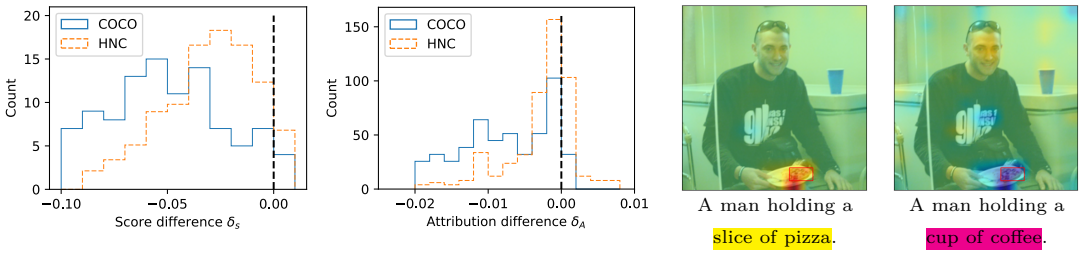


Figure 8: (Left) Histograms for score (δ_S) and attribution (δ_A) changes in hard negative captions. (Right) An example attributions for a hard negative caption, with the true object marked in yellow, and the replaced negative one in magenta. COCO bounding-boxes in red. Details are described in the paragraph *Hard negative captions*.

5 Discussion

Interpretation of results. The fact that the model's intrinsic grounding ability, as measured by our attributions, can be poor on data outside the initial training domain and largely improves upon in-domain tuning points out limitations in the generalization capabilities of CLIP models. Despite the billion scale pre-training, the models seem to require explicit exposure to some object classes to establish a solid correspondence between the vision and language mode for the underlying concepts.

An interesting finding is that CLIP models do not only positively match objects but also penalize mismatches by assigning negative contributions. However, this is not consistently the case and we frequently observe positive cross-attributions between mismatching objects in correlated scenes like tennis courts, bathrooms, kitchens, streets, etc. This suggests that CLIP models may struggle to differentiate between objects that commonly appear together. The contrastive objective may not provide sufficient supervision to learn to tell them apart. Future work should establish a better understanding of this phenomenon.

A solution may be to augment the training data with hard negatives specifically targeting such correlations (Yuksekgonul et al., 2023), potentially making use of synthetic data generation (Patel et al., 2024). Among the failure categories that we have identified, attributions locally exceeding bounding-boxes only imply slight disagreement with human annotations. Wrong attributions due to coverage, actual misidentification, and unusual scenes only account for a small fraction of failures; yet, they may call for more challenging datasets.

Our baseline experiments showed that analyzing interactions in CLIP models is not trivial. Neither simplified gradient-based approaches nor pair-wise embedding multiplication is sufficient for the purpose. Interaction-LIME successfully fits text-side dependencies but cannot reliably capture interactions with image features. In contrast, our approach performs consistently, and requires no optimization or hyper-parameter tuning. Nevertheless, it may be possible to further enhancing our interaction-attributions by incorporating improvements that have been proposed for first-order integrated gradients, including discretization of text representations (Sanyal & Ren, 2021), non-uniform interpolations (Bhat & Raychowdhury, 2023) or integration along non-linear paths (Kapishnikov et al., 2021; Zhuo & Ge, 2024).

Limitations. As stated explicitly in Equation 11, our interaction attributions are an approximation. Throughout this work, we attribute to intermediate representations of inputs, which is both efficient and informative. Attribution to input representations is possible, yet computationally very expensive (Möller et al., 2024). In transformers, intermediate representations have undergone multiple contextualization steps and are technically not strictly tied to input features at a given position. Finally, recently proven fundamental limitations of attribution methods urge caution in their interpretation, especially regarding counterfactual conclusions about the importance of individual features (Bilodeau et al., 2024). Despite these considerations, our empirical evaluations demonstrate that our attributions can effectively point out correspondences between images and captions. Although they should not be regarded as guaranteed robust and faithful explanations, we argue that they offer valuable insights into dual encoder models and have the potential to enhance their development further.

6 Conclusion

In this paper, we derived general feature-pair attributions for dual-encoder architectures, enabling the attribution of similarity predictions onto interactions between input features. Our method is easily applicable to any differentiable dual-encoder architecture and requires no modifications of the initial model. We believe it can provide valuable insights in applications such as (multi-modal) information retrieval and retrieval-augmented generation, helping to identify biases and errors in these models to improve them further.

Applying our method to Clip models provides clear evidence for them capturing fine-grained interactions between corresponding visual and linguistic concepts despite their coarser contrastive objective. Mis-matching objects are often not only ignored but contribute negatively to image-caption similarities. At the same time, we also find pronounced out-of-domain effects, and can identify knowledge gaps about specific object classes in individual models. In-domain fine-tuning can reduce these gaps by large margins, which points out limitations in the initial models' generalization capabilities and complements the recent results by Mayilvahanan et al. (2024). Finally, an error analysis revealed that Clip models can struggle with covered or partially visible objects, unusual scenes, and correlated contexts like kitchens, offices, or sports courts.

By enabling the analysis of interactions between caption and image features, our approach contributes to an emerging interest in understanding higher-order dependencies in Clip models (Gandelsman et al., 2025; Joukovsky et al., 2023), reaching beyond well-understood first-order effects (Zhao et al., 2024).

Acknowledgements

Funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2075 – 390740016. We acknowledge the support by the Stuttgart Center for Simulation Science (SimTech).

References

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, 2022.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, 2015.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 2015.

Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. Effective conditioned and composed image retrieval combining clip-based features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21466–21474, June 2022.

Jasmijn Bastings and Katja Filippova. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 2020.

Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio Calmon, and Himabindu Lakkaraju. Interpreting clip with sparse linear concept embeddings (splice). *Advances in Neural Information Processing Systems*, 2025.

Ashwin Bhat and Arijit Raychowdhury. Non-uniform interpolation in integrated gradients for low-latency explainable-ai. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2023. doi: 10.1109/ISCAS46773.2023.10181829.

Blair Bilodeau, Natasha Jaques, Pang Wei Koh, and Been Kim. Impossibility theorems for feature attribution. *Proceedings of the National Academy of Sciences*, 2024.

Samuel Black, Abby Stylianou, Robert Pless, and Richard Souvenir. Visualizing paired image similarity in transformer networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022.

Walid Bousselham, Angie Boggust, Sofian Chaybouti, Hendrik Strobelt, and Hilde Kuehne. Legrad: An explainability method for vision transformers via feature formation sensitivity. *arXiv preprint arXiv:2404.03214*, 2024.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 2020.

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021.

Peijie Chen, Qi Li, Saad Biaz, Trung Bui, and Anh Nguyen. gscorecam: What objects is clip looking at? In *Proceedings of the Asian Conference on Computer Vision*, 2022.

Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish V Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme Ruiz, Andreas Peter Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. PaLI: A jointly-scaled multilingual language-image model. In *The Eleventh International Conference on Learning Representations*, 2023a.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, 2020.

Zhihong Chen, Ruifei Zhang, Yibing Song, Xiang Wan, and Guanbin Li. Advancing visual grounding with scene knowledge: Benchmark and method. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023b.

Tianyu Cui, Pekka Marttinen, and Samuel Kaski. Learning global pairwise interactions with bayesian neural networks. In *ECAI 2020*, pp. 1087–1094. IOS Press, 2020.

Samyak Datta, Karan Sikka, Anirban Roy, Karuna Ahuja, Devi Parikh, and Ajay Divakaran. Align2ground: Weakly supervised phrase grounding guided by image-caption alignment. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2019.

Eustasio del Barrio, Juan A. Cuesta-Albertos, and Carlos Matrán. *An Optimal Transportation Approach for Assessing Almost Stochastic Order*, pp. 33–44. Springer, 2018.

Esra Dönmez, Pascal Tilli, Hsiu-Yu Yang, Ngoc Thang Vu, and Carina Silberer. Hnc: Leveraging hard negative captions towards models with fine-grained visual-linguistic comprehension capabilities. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, 2023.

Michael Dorkenwald, Nimrod Barazani, Cees GM Snoek, and Yuki M Asano. Pin: Positional insert unlocks object localisation abilities in vlms. *arXiv preprint arXiv:2402.08657*, 2024.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Rotem Dror, Segev Shlomov, and Roi Reichart. Deep dominance - how to properly compare deep neural models. In *Proceedings of the 57th ACL*, 2019.

Oliver Eberle, Jochen Büttner, Florian Kräutli, Klaus-Robert Müller, Matteo Valleriani, and Grégoire Montavon. Building and interpreting deep similarity models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. Improving clip training with language rewrites. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, 2023.

Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander T Toshev, and Vaishaal Shankar. Data filtering networks. In *The Twelfth International Conference on Learning Representations*, 2024.

Fabian Fumagalli, Maximilian Muschalik, Patrick Kolpaczki, Eyke Hüllermeier, and Barbara Hammer. Shapiq: Unified approximation of any-order shapley interactions. *Advances in Neural Information Processing Systems*, 2024.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah M Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. Interpreting clip's image representation via text-based decomposition. *arXiv preprint arXiv:2310.05916*, 2023.

Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. Interpreting the second-order effects of neurons in CLIP. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xingyu Gao, Steven CH Hoi, Yongdong Zhang, Ji Wan, and Jintao Li. Soml: Sparse online metric learning with application to image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2014.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *ArXiv*, 2023.

Loris Giulivi and Giacomo Boracchi. Concept visualization: Explaining the clip multi-modal embedding using wordnet. In *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024.

Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan Rossi, Vishwa Vinay, and Aditya Grover. Cyclip: Cyclic contrastive language-image pretraining. In *Advances in Neural Information Processing Systems*, 2022.

Michel Grabisch and Marc Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of game theory*, 1999.

Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Is that you? metric learning approaches for face identification. In *2009 IEEE 12th international conference on computer vision*, 2009.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, 2016.

Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding visual explanations. In *Proceedings of the European conference on computer vision (ECCV)*, 2018.

Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *Similarity-Based Pattern Recognition: Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3*, 2015.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems*, 2019.

Sarthak Jain and Byron C. Wallace. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

Joseph D Janizek, Pascal Sturmfels, and Su-In Lee. Explaining explanations: Axiomatic feature interactions for deep networks. *Journal of Machine Learning Research*, 2021.

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 *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Boris Joukovsky, Fawaz Sammani, and Nikos Deligiannis. Model-agnostic visual explanations via approximate bilinear models. In *2023 IEEE International Conference on Image Processing (ICIP)*, 2023.

Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, and Tolga Bolukbasi. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5050–5058, June 2021.

Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 2019.

Rémi Kazmierczak, Eloïse Berthier, Goran Frehse, and Gianni Franchi. CLIP-QDA: An explainable concept bottleneck model. *Transactions on Machine Learning Research*, 2024.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 2020.

Jing Yu Koh, Ruslan Salakhutdinov, and Daniel Fried. Grounding language models to images for multimodal inputs and outputs. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Janghyeon Lee, Jongsuk Kim, Hyounguk Shon, Bumsoo Kim, Seung Hwan Kim, Honglak Lee, and Junmo Kim. Uniclip: Unified framework for contrastive language-image pre-training. In *Advances in Neural Information Processing Systems*, 2022.

Samuel Lerman, Charles Venuto, Henry Kautz, and Chenliang Xu. Explaining local, global, and higher-order interactions in deep learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1224–1233, 2021.

Martha Lewis, Nihal Nayak, Peilin Yu, Jack Merullo, Qinan Yu, Stephen Bach, and Ellie Pavlick. Does CLIP bind concepts? probing compositionality in large image models. In Yvette Graham and Matthew Purver (eds.), *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 1487–1500, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL `https://aclanthology.org/2024.findings-eacl.101/`.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the AAAI conference on artificial intelligence*, 2020.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *Advances in Neural Information Processing Systems*, 2021.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the 39th International Conference on Machine Learning*, 2022a.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022b.

Yi Li, Hualiang Wang, Yiqun Duan, Hang Xu, and Xiaomeng Li. Exploring visual interpretability for contrastive language-image pre-training. *arXiv preprint arXiv:2209.07046*, 2022c.

Yi Li, Hualiang Wang, Yiqun Duan, and Xiaomeng Li. Clip surgery for better explainability with enhancement in open-vocabulary tasks. *arXiv preprint arXiv:2304.05653*, 2023.

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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 2014.

Zachary C. Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 2018.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. UNIFIED-IO: A unified model for vision, language, and multi-modal tasks. In *The Eleventh International Conference on Learning Representations*, 2023.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, 2017.

Joanna Materzyńska, Antonio Torralba, and David Bau. Disentangling visual and written concepts in clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Prasanna Mayilvahanan, Roland S Zimmermann, Thaddäus Wiedemer, Evgenia Rusak, Attila Juhos, Matthias Bethge, and Wieland Brendel. In search of forgotten domain generalization. *arXiv preprint arXiv:2410.08258*, 2024.

Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. *Layer-Wise Relevance Propagation: An Overview*, pp. 193–209. Springer, 2019.

Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. SLIP: Self-supervision meets language-image pre-training. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, 2022.

W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 2019.

Basil Mustafa, Carlos Riquelme, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby. Multimodal contrastive learning with limoe: the language-image mixture of experts. In *Advances in Neural Information Processing Systems*, 2022.

Lucas Möller, Dmitry Nikolaev, and Sebastian Padó. An attribution method for siamese encoders. In *Proceedings of EMNLP*, 2023.

Lucas Möller, Dmitry Nikolaev, and Sebastian Padó. Approximate attributions for off-the-shelf Siamese transformers. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024.

Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Maitreya Patel, Naga Sai Abhiram Kusumba, Sheng Cheng, Changhoon Kim, Tejas Gokhale, Chitta Baral, et al. Tripletclip: Improving compositional reasoning of clip via synthetic vision-language negatives. *Advances in neural information processing systems*, 37:32731–32760, 2024.

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 (ICCV)*, 2015.

Bryan A. Plummer, Mariya I. Vasileva, Vitali Petsiuk, Kate Saenko, and David Forsyth. Why do these match? explaining the behavior of image similarity models. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI*, 2020.

Shraman Pramanick, Li Jing, Sayan Nag, Jiachen Zhu, Hardik J Shah, Yann LeCun, and Rama Chellappa. VoLTA: Vision-language transformer with weakly-supervised local-feature alignment. *Transactions on Machine Learning Research*, 2023.

Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Vincent Quantmeyer, Pablo Mosteiro, and Albert Gatt. How and where does CLIP process negation? In *Proceedings of the 3rd Workshop on Advances in Language and Vision Research (ALVR)*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Karthikeyan Natesan Ramamurthy, Amit Dhurandhar, Dennis Wei, and Zaid Bin Tariq. Analogies and feature attributions for model agnostic explanation of similarity learners. *arXiv preprint arXiv:2202.01153*, 2022.

Ali Rasekh, Sepehr Kazemi Ranjbar, Milad Heidari, and Wolfgang Nejdl. Ecor: Explainable clip for object recognition. *arXiv preprint arXiv:2404.12839*, 2024.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Bjorn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *International Conference on Machine Learning*, 2020.

Arka Sadhu, Kan Chen, and Ram Nevatia. Zero-shot grounding of objects from natural language queries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Dylan Sam, Devin Willmott, Joao D Semedo, and J Zico Kolter. Finetuning clip to reason about pairwise differences. *arXiv preprint arXiv:2409.09721*, 2024.

Fawaz Sammani, Boris Joukovsky, and Nikos Deligiannis. Visualizing and understanding contrastive learning. *IEEE Transactions on Image Processing*, 2023.

Soumya Sanyal and Xiang Ren. Discretized integrated gradients for explaining language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 10285–10299, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.805. URL `https://aclanthology.org/2021.emnlp-main.805/`.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems*, 2022.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? In *International Conference on Learning Representations*, 2021.

K Simonyan, A Vedaldi, and A Zisserman. Deep inside convolutional networks: visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations (ICLR)*. ICLR, 2014.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, 2016.

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 *International Conference on Learning Representations*, 2019.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. The shapley taylor interaction index. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Pascal Tilli and Ngoc Thang Vu. Discrete subgraph sampling for interpretable graph based visual question answering. In *Proceedings of the 31st International Conference on Computational Linguistics*, 2025.

Michael Tsang, Dehua Cheng, and Yan Liu. Detecting statistical interactions from neural network weights. In *International Conference on Learning Representations*, 2018.

Weijie Tu, Weijian Deng, and Tom Gedeon. A closer look at the robustness of contrastive language-image pre-training (clip). *Advances in Neural Information Processing Systems*, 2024.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2019.

Alexandros Vasileiou and Oliver Eberle. Explaining text similarity in transformer models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024.

Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020.

Ying Wang, Tim GJ Rudner, and Andrew G Wilson. Visual explanations of image-text representations via multi-modal information bottleneck attribution. *Advances in Neural Information Processing Systems*, 2023.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *2018 IEEE winter conference on applications of computer vision (WACV)*, 2018.

C. Xie, S. Sun, X. Xiong, Y. Zheng, D. Zhao, and J. Zhou. Ra-clip: Retrieval augmented contrastive language-image pre-training. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=zeFrfgyZln`.

Hu Xu, Saining Xie, Xiaoqing Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying CLIP data. In *The Twelfth International Conference on Learning Representations*, 2024.

J. Yang, J. Duan, S. Tran, Y. Xu, S. Chanda, L. Chen, B. Zeng, T. Chilimbi, and J. Huang. Vision-language pre-training with triple contrastive learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022a.

J. Yang, C. Li, P. Zhang, B. Xiao, C. Liu, L. Yuan, and J. Gao. Unified contrastive learning in image-text-label space. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022b.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. Unitab: Unifying text and box outputs for grounded vision-language modeling. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVI*, 2022c.

Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

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*, 2014.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, 2022.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=KRLUvxh8uaX`.

Andrew Zhai and Hao-Yu Wu. Classification is a strong baseline for deep metric learning. *arXiv preprint arXiv:1811.12649*, 2018.

Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 2018.

Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European conference on computer vision*, 2022a.

Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. Contrastive learning of medical visual representations from paired images and text. In *Machine Learning for Healthcare Conference*, 2022b.

Chenyang Zhao and Antoni B. Chan. ODAM: Gradient-based instance-specific visual explanations for object detection. In *The Eleventh International Conference on Learning Representations*, 2023.

Chenyang Zhao, Kun Wang, Xingyu Zeng, Rui Zhao, and Antoni B. Chan. Gradient-based visual explanation for transformer-based CLIP. In *Forty-first International Conference on Machine Learning*, 2024.

Meng Zheng, Srikrishna Karanam, Terrence Chen, Richard J Radke, and Ziyan Wu. Towards visually explaining similarity models. *arXiv preprint arXiv:2008.06035*, 2020.

J. Zhou, L. Dong, Z. Gan, L. Wang, and F. Wei. Non-contrastive learning meets language-image pre-training. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. Large language models for information retrieval: A survey, 2024. URL `https://arxiv.org/abs/2308.07107`.

Yue Zhuo and Zhiqiang Ge. Ig2: Integrated gradient on iterative gradient path for feature attribution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(11):7173–7190, 2024. doi: 10.1109/TPAMI.2024.3388092.

(a) A couple sitting on a bench looking at the sea. (b) A couple sitting on a bench looking at the sea. (c) A dog is jumping for a frisbee. (d) A dog is jumping for a frisbee.

Figure 9: Additional examples for inter-modal attributions of token-range selection with image projections (left) and bounding-box selection with caption projection (right). The visualization is identical to Figure 1.

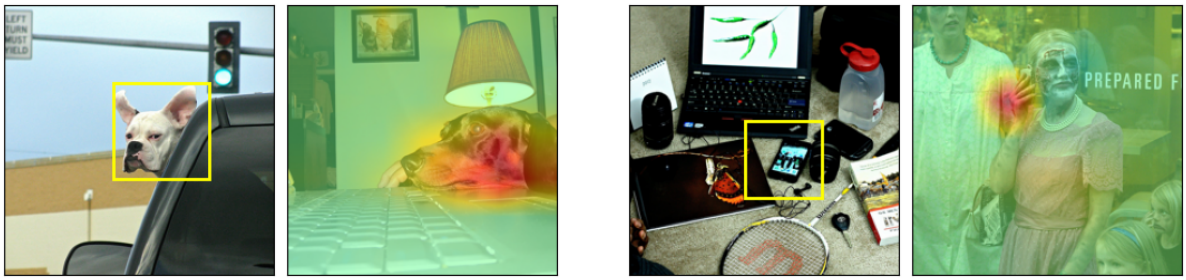


Figure 10: Image-image attributions between the yellow bounding-box in the left image and the one to its right as described in Section 3. Visualisation is identical to Figure 2 (right)

A Additional Examples

Figure 9 shows two more examples for inter-modal attributions, one for text-span selection and image projection and one for bounding-box selection and caption projection.

Figure 10 includes two more examples for image-image attributions as described in Section 3 under *intra-modal attributions*.

Figure 11 shows a qualitative comparison between our attributions and the baselines described in Section 4.1.

Figure 12 shows five examples for each of the five failure categories that we identified in Section 4.2 under *Qualitative failure analysis*.

B Extended results

Table 3 shows full results for our Point-Game evaluation on different OpenAI models. Next to the ViT-B-16 architecture, we also evaluate the RN50 and ViT-B-32 variants. Table 4 includes the full evaluation for all OpenClip models. In addtion to the median PGE (mPGE), in these tables we also report cumulative PGE densitites for the 80th percentile (PGE>0.8). Full cumulative PGE-histograms for additional models are included in Figures 13 and 14.

Table 5 presents full results of our Point-Game baseline experiments extending Section 4.1. Corresponding cumulative densities of the PGE-metric are shown in Figure 15. Figures 16 and 17 show the plots of the conditional insertion and deletion experiments for the OpenClip Laion model and the original OpenAI model, respectively. The corresponding AUC values are contained in Table 1.

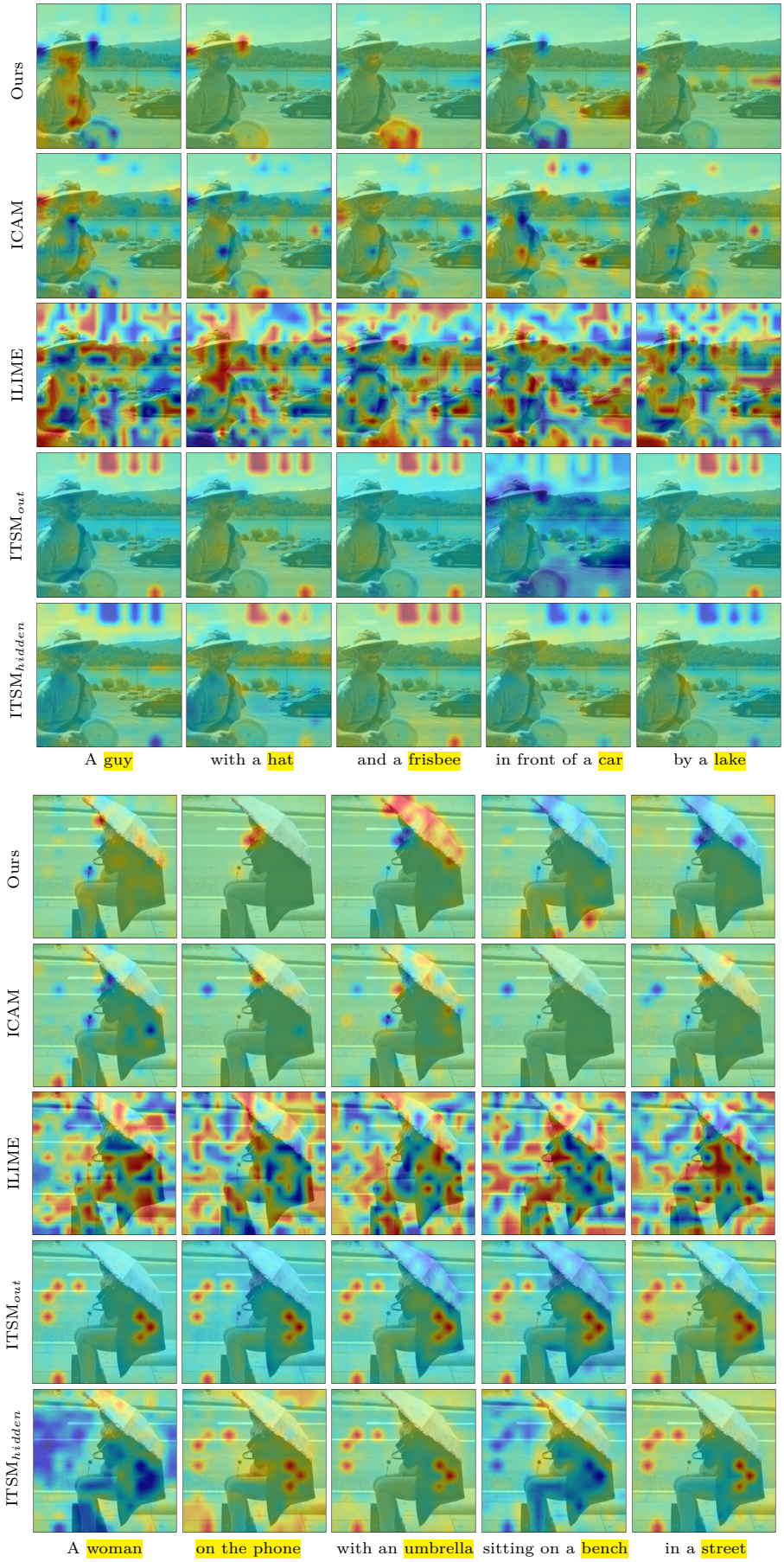


Figure 11: Qualitative comparison between our attributions, the InteractionCAM (ICAM), InteractionLIME (ILIME) and both ITSM variants. Heatmaps over images in a given column are for the marked parts of the captions in yellow below.

Figure 12: Five examples for each of the five identified failure categories as described in Section 4.2.

		Coco			HNC			Flickr30k		
Model	**Tuning**	mPGE	PGE>0.8	PGA	mPGE	PGE>0.8	PGA	mPGE	PGE>0.8	PGA
RN50	No	66.3	28.8	76.9	50.1	22.6	61.8	60.1	25.5	71.2
ViT-B/32	No	63.5	33.3	69.1	52.8	28.5	58.5	50.4	23.4	58.1
ViT-B/16	No	72.3	35.7	79.0	57.0	31.7	65.0	64.4	28.4	72.1
ViT-B-16	Yes	**78.0**	**48.4**	**82.9**	-	-	-	**73.4**	**40.7**	**79.0**

Table 3: Summary of Point-Game evaluation for different Clip models by OpenAI as described in Section 4.1. *Model* refers to the investigated architecture, *Tuning* is whether the model was fine-tuned on the train split of the respective dataset. Best overall results are in bold, best results of unmodified models are underlined.

		COCO			Flickr30k		
Training	**Tuning**	mPGE	PGE>0.8	PGA	mPGE	PGE>0.8	PGA
Laion	No	49.4	22.0	63.3	38.2	15.9	52.0
	Yes	71.1	47.3	**83.2**	**54.6**	30.6	**61.8**
CommonPool	No	43.0	18.2	58.8	36.7	15.5	53.0
	Yes	57.7	28.7	67.1	44.6	20.8	56.2
DataComp	No	38.5	14.6	56.0	32.8	11.8	48.9
	Yes	**72.4**	50.0	75.1	50.7	27.3	56.0
DFN	No	46.5	19.6	54.3	35.4	12.3	43.3
	Yes	71.4	**53.3**	74.6	53.1	**33.5**	58.3
Meta-Clip	No	44.2	16.8	52.3	37.0	14.5	46.4
	Yes	57.5	49.8	77.1	49.2	24.1	57.2

Table 4: Summary of the Point-Game evaluation for all OpenClip models on Coco and Flickr30k. The *Training* column refers to the dataset the model was initially trained on, *Tuning* is whether the model was additionally fine-tuned on the train-split of the respective evaluation dataset. All models implement the ViT-B-16 architecture except Meta-Clip that uses quickgelu activations. Best overall results are in bold, best results for unmodified models are underlined.

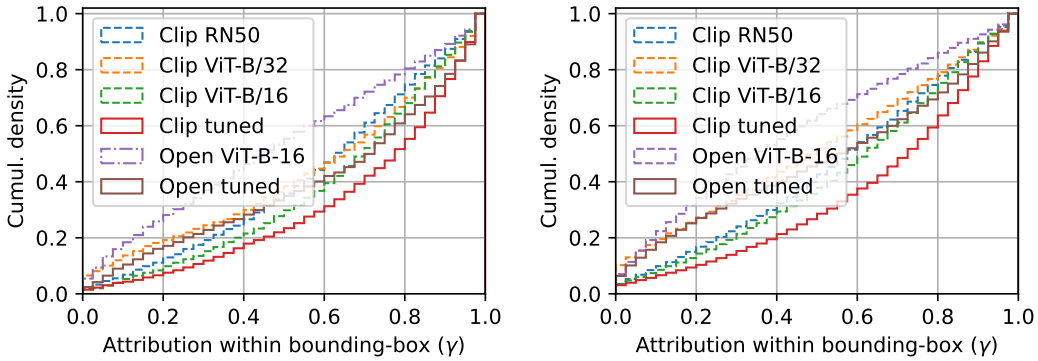


Figure 13: Cumulative PGE-distribution plots of the OpenAI models for the Coco (left) and Flickr30k (right) dataset as described in Section 4.1.

Figure 18 extends the class-wise PGE-evaluation from Section 4.2 to the OpenClip Dfn and DataComp models.

		Coco		Flickr30k	
Training	**Method**	mPGE	PGA	mPGE	PGA
OpenAI	ITSM_{out}	18.1	21.4	19.5	23.3
	$\text{ITSM}_{\text{hidden}}$	29.8	38.1	29.5	37.4
	ILIME	27.9	34.9	25.8	33.1
	ICAM	38.6	54.6	33.5	51.4
	Ours	**72.3**	**79.0**	**64.4**	**72.1**
Laion (tuned)	ITSM_{out}	22.8	30.3	24.5	28.7
	$\text{ITSM}_{\text{hidden}}$	30.5	34.6	28.8	36.6
	ILIME	28.8	37.8	25.8	34.5
	ICAM	32.5	58.4	33.5	51.4
	Ours	**71.2**	**83.2**	**56.3**	**63.6**
Dfn (tuned)	ITSM_{out}	24.2	34.5	25.1	31.4
	$\text{ITSM}_{\text{hidden}}$	27.4	23.0	27.7	36.5
	ILIME	27.9	39.2	25.7	33.5
	ICAM	33.3	46.5	24.2	42.2
	Ours	**71.4**	**74.6**	**53.1**	**58.3**
DataComp (tuned)	ITSM_{out}	25.5	38.7	26.5	33.9
	$\text{ITSM}_{\text{hidden}}$	35.0	42.3	22.6	28.4
	ILIME	28.4	39.3	25.7	34.1
	ICAM	36.9	49.5	23.2	37.3
	Ours	**72.4**	**75.1**	**50.7**	**60.0**

Table 5: PGE-evaluation results of our method compared against the ITSM and InteractionCAM (ICAM) baselines for different models as described in Section 4.1 under *Object localization*. Best results for every model are in bold.

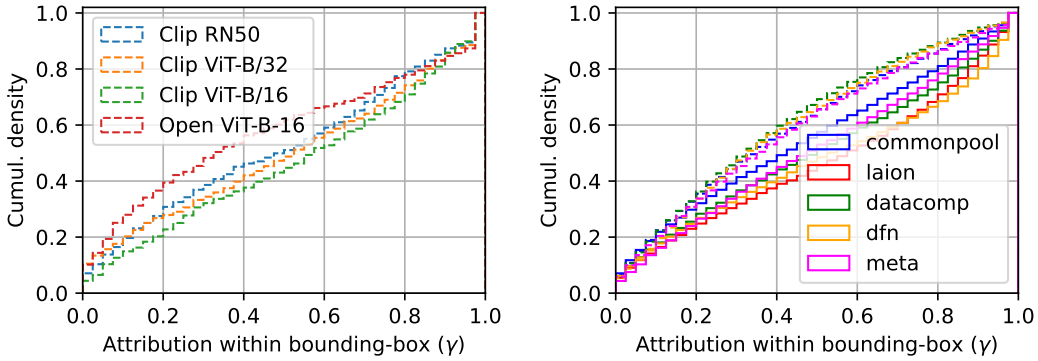


Figure 14: Cumulative PGE-distribution plots for the OpenAI models on HNC (left) and the OpenClip models on Flickr30k (right) as described in Section 4.1.

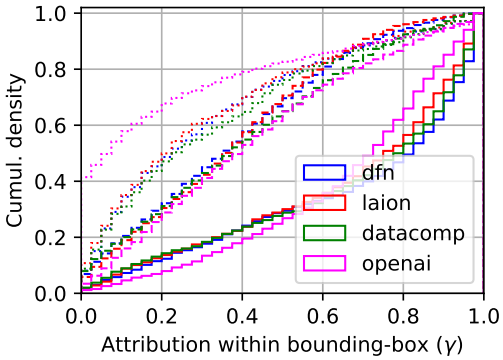
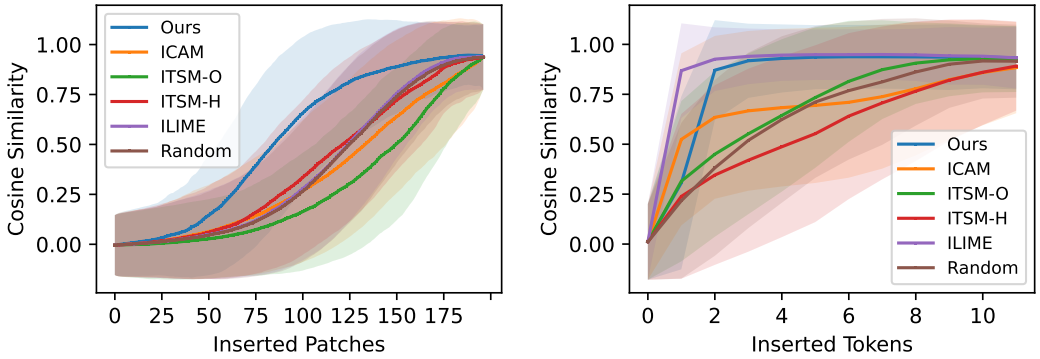


Figure 15: Cumulative PGE-distributions for our baseline experiment in Section 4.1. Our method is in solid, InteractionCAM is dashed and ITSM_{out} is dotted. ITSM_{hidden} is excluded for an uncluttered visualization.

C Additional Experiments

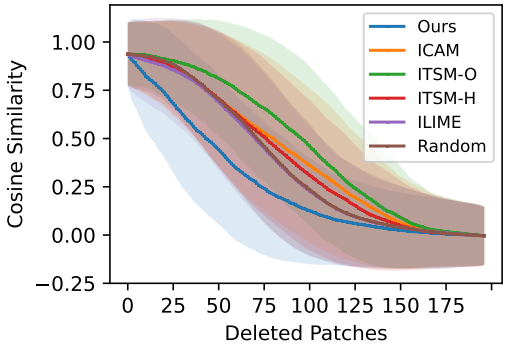
Approximation Error. In Section 3, we have shown the equality between Eq. 2 and Eq. 10. The only approximation affecting this equality is the numerical integration in Eq. 9 to calculate the two *integrated Jacobians* $\mathbf{J}^a$ and $\mathbf{J}^b$ by a sum over N bins. We can evaluate how good this approximation is by explicitly calculating the four similarity predictions between the references and inputs $f(\mathbf{a}, \mathbf{b})$, $f(\mathbf{r}_a, \mathbf{b})$, $f(\mathbf{a}, \mathbf{r}_b)$ and $f(\mathbf{r}_a, \mathbf{r}_b)$, as well as the attribution matrix $\mathbf{A}$. The *approximation error* can then be defined as the absolute difference between Eq. 2 and Eq. 10. In Figure 19, we plot this error as a function of different magnitudes for N. For larger N, it converges as expected due to the Riemann definition of the integral.

Choice of references. The choice of the references $\mathbf{r}_a$ and $\mathbf{r}_b$ is ambiguous as long as they are uninformative. We try different options and evaluate their approximation errors as defined above. For the image input, we use a black image and zero-centered gaussian noise. For the text input, we use a sequence of padding tokens and the empty sequence consisting only of the `CLS` and `EOS` tokens. Figure 19 includes approximation errors for all four combinations of these references as a function of N. Combinations with gaussian noise for the image reference appear to converge slightly faster than the black image.
For different references, attributions can vary slightly. However, these differences are small, even for large objects like the cow in Figure 20 (left), for which attributions tend to spread out. The absolute difference of attributions between any two combinations of references is on the order of 10^{-5} and tends to decrease for larger N. The plot in Figure 20 (right) shows this for the difference between a *black-padding* and *gaussian-empty* reference combination.

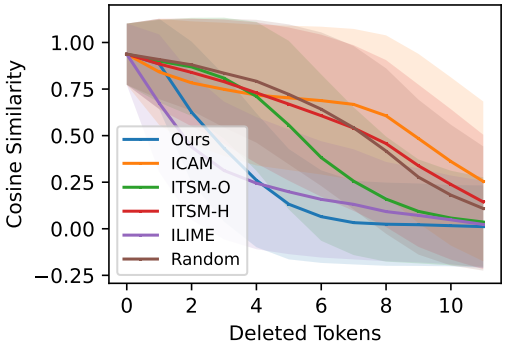


(a) Conditional Image Insertion.

(b) Conditional Text Insertion.



(c) Conditional Image Deletion.



(d) Conditional Text Deletion.

Figure 16: Average change in similarity score upon conditional insertion and deletion performed on either the caption or the image using a ViT-B-16 model pretrained on LAION. Confidence intervals are standard deviations over the evaluation dataset. Table 1 summarizes the AUC of these plots.

D Alternative Figure 1

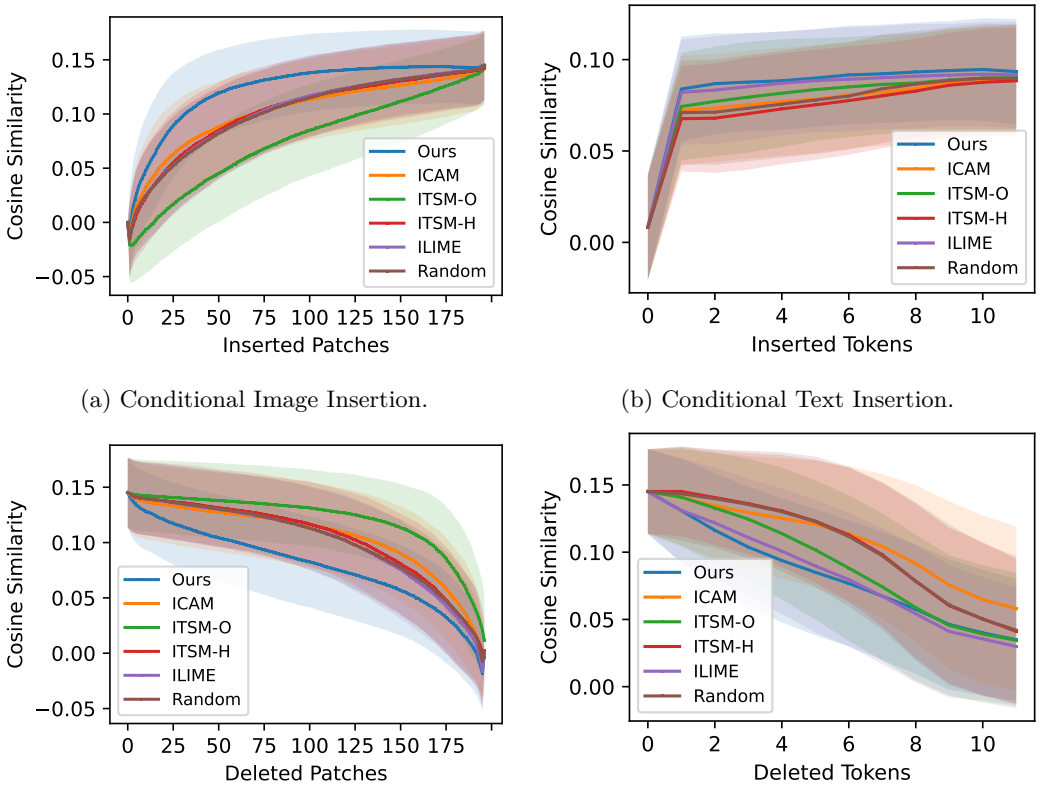
E Stochastic Dominance

Stochastic dominance defines an order relation between probability distributions based on their cumulatives. del Barrio et al. (2018) have proposed a significance test building on the principle and Dror et al. (2019) have identified it as being particularly suitable to compare deep neural models. The test's ϵ-parameter is the maximal percentile range where the inferior distribution is allowed to dominate the superior one and Dror *et al.* suggest to set it to $\epsilon < 0.4$. The smaller ϵ, the stricter the criterion. α is the significance level.

F Relation to interactionCAM

Here, we first discuss the relation of integrated gradients Sundararajan et al. (2017) and GradCam and then show how our method can be reduced to the Interaction-CAM baseline through simplification. We start by deriving IG for a model $f(\mathbf{a}) = s$ with a vector-valued input $\mathbf{a}$ and a scalar prediction s, which might e.g. be a classification score. We define the reference input $\mathbf{r}$, begin from the difference between the two predictions and reformulate it as an integral over the integration variable $\mathbf{x}$:

$$f(\mathbf{a}) - f(\mathbf{r}) = \int_{\mathbf{r}}^{\mathbf{a}} \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i} d\mathbf{x}_i \tag{12}$$



(a) Conditional Image Insertion.

(b) Conditional Text Insertion.

(c) Conditional Image Deletion.

(d) Conditional Text Deletion.

Figure 17: Conditional insertion and deletion performed on either the caption or the image using the original ViT-B-16 model by OpenAI without any fine-tuning.

Again we do not write out sums over double indices. To solve the resulting line integral, we substitute with the straight line $\mathbf{x}(\alpha) = \mathbf{r} + \alpha(\mathbf{a} - \mathbf{r})$ and pull its derivative $d\mathbf{x}(\alpha)/d\alpha = (\mathbf{a} - \mathbf{r})$ out of the integral:

$$\int_{\alpha=0}^{1} \frac{\partial f(\mathbf{x}(\alpha))}{\partial \mathbf{x}_i(\alpha)} \frac{\partial \mathbf{x}_i(\alpha)}{\partial \alpha} d\alpha = (\mathbf{a} - \mathbf{r})_i \int_{\alpha=1}^{1} \nabla_i f(\mathbf{x}(\alpha))\, d\alpha \tag{13}$$

In practice, we approximate the integral by a sum over N steps. If the reference is uninformative, so that $f(\mathbf{r}) \approx 0$, the equality between Eq. 12 and Eq. 13 can be reduced to the final approximation of IG:

$$f(\mathbf{a}) \approx (\mathbf{a} - \mathbf{r})_i \frac{1}{N} \sum_{n=1}^{N} \nabla_i f(\mathbf{x}(\alpha_n)), \tag{14}$$

which decomposes the model prediction $f(\mathbf{a})$ into contributions of individual feature i in $\mathbf{a}$. We can now reduce these feature attributions further by setting $N = 1$ and $\mathbf{r} = \mathbf{0}$, to obtain

$$\mathbf{a}_i \nabla_i f(\mathbf{a}), \tag{15}$$

which is often referred to as *gradient×input* and is the basic form of GradCam. The method typically attributes to deep image representations in CNNs, so that $\mathbf{a}$ has the dimensions $C \times H \times W$, the number of channels, height and width of the representation. To reduce attributions to a two-dimensional map, it sums over the channel dimension and applies a relu-activation to the outcome. The original version also average pools the gradients over the spacial dimensions, however, this is technically not necessary.

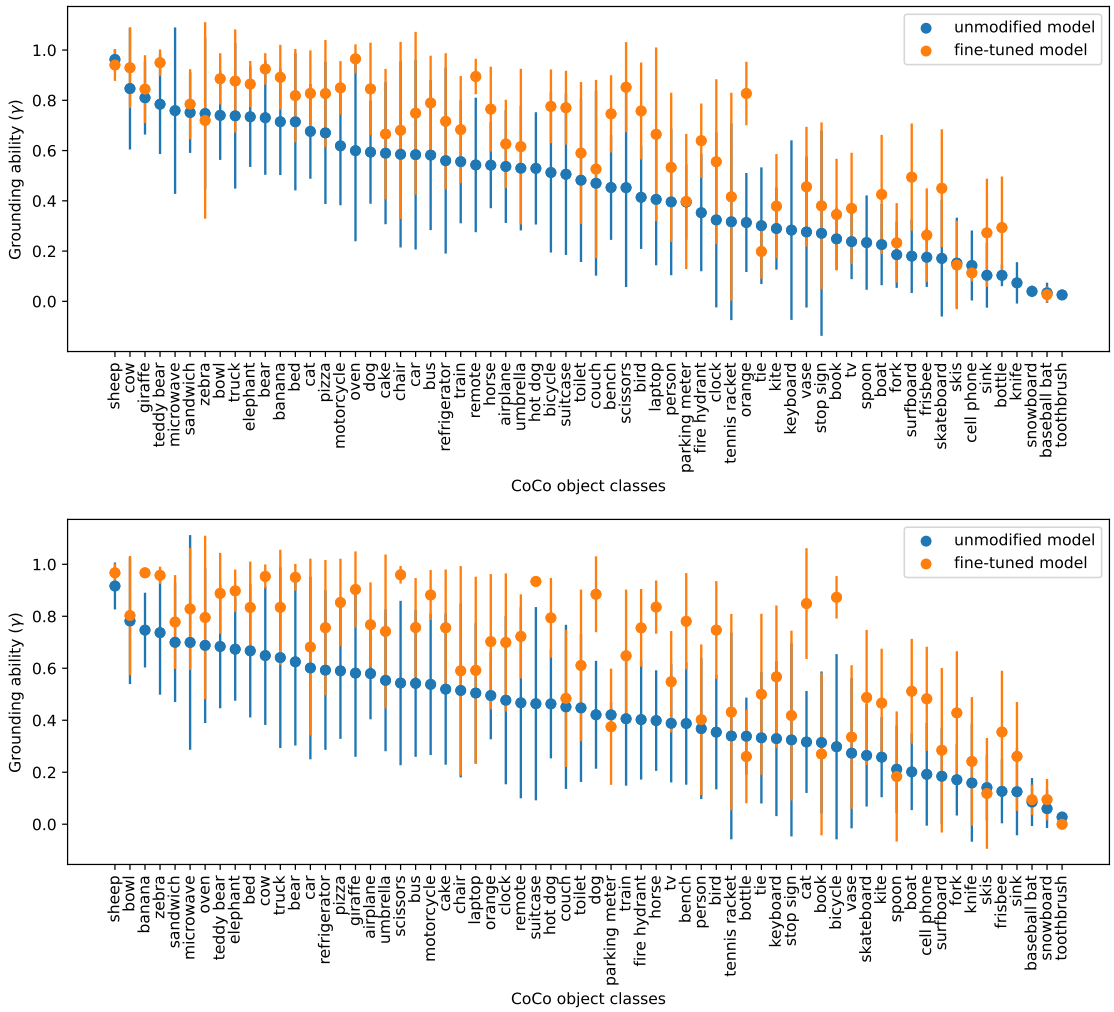


Figure 18: Class-wise PGE-evaluation for the OpenClip Laion (top) and DataComp (bottom) models before and after in-domain fine-tuning as discussed in Section 4.1.

As discussed earlier, neither integrated gradients nor GradCam can explain dual encoder predictions. Following the logic from above we can, however, reduce our feature-interaction attributions from Eq. 10 by setting $N = 1$ in the computation of the integrated Jacobians in Eq. 9 and using $\mathbf{r}_a = \mathbf{r}_b = \mathbf{0}$. For our attribution matrix from Equation 10 we then receive the simplified version

$$\mathbf{a}_i \frac{\partial \mathbf{g}_k}{\partial \mathbf{a}_i} \frac{\partial \mathbf{h}_k}{\partial \mathbf{b}_j} \mathbf{b}_j. \tag{16}$$

This simplification could be termed *Jacobians×inputs* and is equivalent to the Interaction-CAM by Sammani et al. (2023). Note, however, that setting $N = 1$ is the worst possible approximation to the integrated Jacobians. Therefore, it is not surprising that empirically this version performs worse than our full attributions.

G Interaction LIME

We reimplement the InteractionLIME method proposed by Joukovsky et al. (2023) that extends the principle of LIME Ribeiro et al. (2016) to dual encoder models with two inputs.
The core idea of LIME is to locally approximate the actual model f around a given input with an interpretable

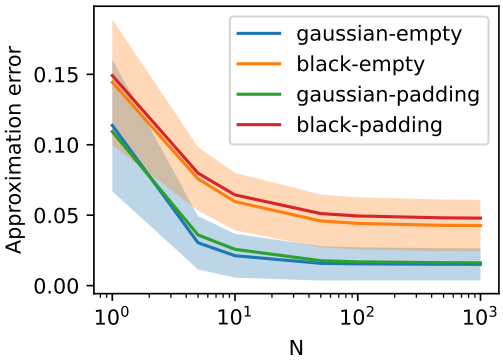


Figure 19: Approximation errors for different reference choices as a function of the number of integration steps N. The image references are abbreviated as 'gaussian' for gaussian noise and 'black' for the black image. Text references are 'padding' and 'empty' for a padding sequence and the empty sequence, respectively. Exemplatory standard deviations over the evaluation sample are shown as shades of the respective plots.

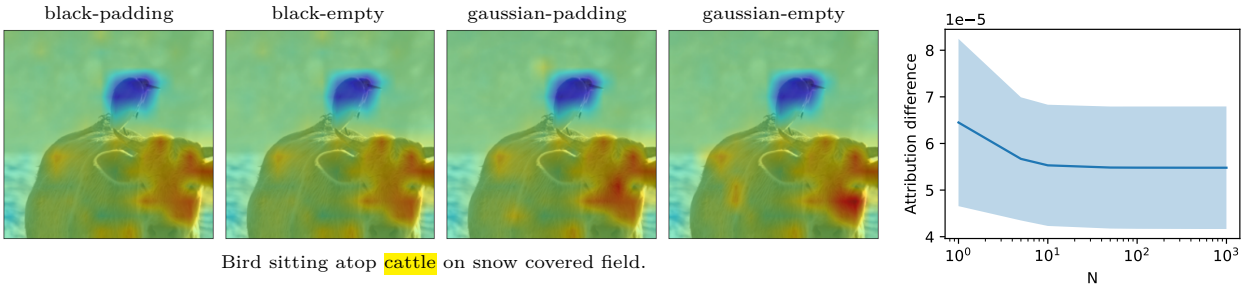


Figure 20: (Left) Attribution differences for all four combinations of references as indicated above the images for the yellow selection in the caption below. (Right) The mean absolute attribution difference between a *black-padding* and a *gaussian-empty* reference combination as a function of the number of approximation steps N and its standard deviation over the evaluation smaple (blue shade).

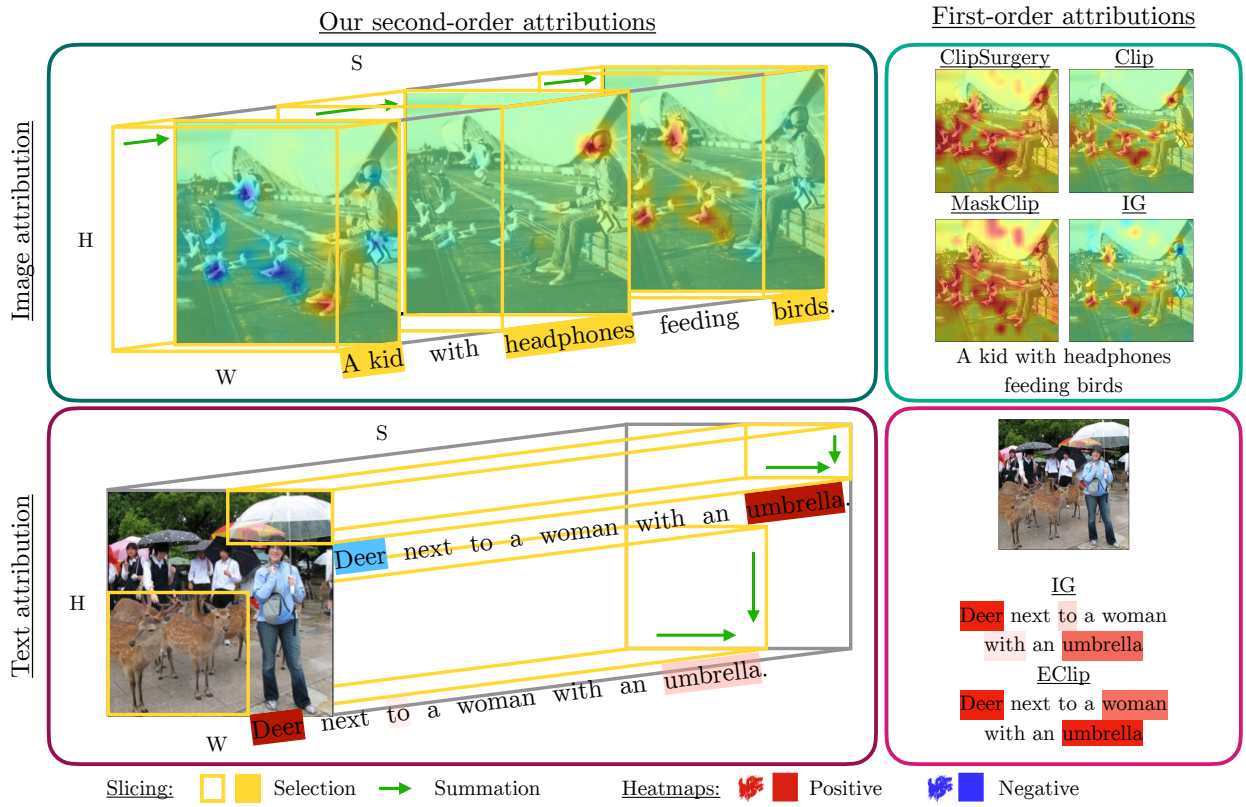


Figure 21: Alternative illustration of the three dimensional character of our interaction attributions (left column) vs. first-order feature attributions (right column). Our attributions are a 3d volume with the image dimensions (H, W) and the sequential text dimension S. Every entry in this volume corresponds to the interaction between the given token and image patch at this position. We make a selection over a slice of this volumn (yellow boxes) and project it onto one of the input modes by summation along the other dimensions (green arrows). This either results in heatmaps over the image dimensions for a text-span selection (top-left) or a heatmap over the text dimension (bottom-left) for a selection of an image region.

surrogate model φ. The local neighborhood of the input is approximated by a sample of perturbations. The surrogate model is typically linear and operates on latent representations of the input. Further, there needs to be a mapping from latent representations to input representations, so that we can generate corresponding inputs that the actual model can process.

In the image domain latent representations $\mathbf{z}^a$ are typically binary variables indicating the presence or absence of super pixels in the input. To enable a direct comparison to our method and the other baselines, we use the vision transformer's patches as super pixels. Analogously, in the text input we define latent representations $\mathbf{z}^b$ as binary variables indicating the presence of input tokens. Disabled image patches are replaced with the mean over the image, disabled tokes are replaced with the padding token.

The local neighborhood of a given input pair $(\mathbf{a}, \mathbf{b})$ is approximated by sampling N such latent representations $(\mathbf{z}_i^a, \mathbf{z}_i^b)$ from two Bernoulli distributions. For the corresponding input perturbations $(\mathbf{a}_i, \mathbf{b}_i)$, we then compute the CLIP scores $s_i = f(\mathbf{a}_i, \mathbf{b}_i)$ and fit the surrogate model to reproduce these predictions.

To account for interactions between the two inputs in dual encoder models, Joukovsky et al. (2023) propose to use a bilinear form as surrogate model:

$$\varphi(\mathbf{z}^a, \mathbf{z}^b) = \mathbf{z}^{a\top} \mathbf{W} \mathbf{z}^b + c, \tag{17}$$

with a weight matrix $\mathbf{W}$ and a scalar bias c, which is then optimized according to the following MSE objective:

$$\min_{\mathbf{W},c} \sum_{i=1}^{N} \pi(\mathbf{a}, \mathbf{a}_i, \mathbf{b}, \mathbf{b}_i) \Big(f(\mathbf{a}_i, \mathbf{b}_i) - \varphi(\mathbf{z}_i^a, \mathbf{z}_i^b; \mathbf{W}, c) \Big)^2 \tag{18}$$

Here, π is a function that weights individual neighborhood samples $(\mathbf{a}_i, \mathbf{b}_i)$ according to their similarity to the original input $(\mathbf{a}, \mathbf{b})$. We use the cosine similarities between perturbed and original captions and image inputs, respectively, and following Joukovsky et al. (2023), define the total similarity weight as the average of the caption and image similarity:

$$\pi(\mathbf{a}, \mathbf{a}_i, \mathbf{b}, \mathbf{b}_i) = \frac{1}{2} \big(\mathbf{g}^\top(\mathbf{a})\, \mathbf{g}(\mathbf{a}_i) + \mathbf{h}^\top(\mathbf{b})\, \mathbf{h}(\mathbf{b}_i) \big) \tag{19}$$

To fit φ, we use stochastic gradient descent with a learning rate of 10^{-2} and weight-decay of 10^{-3} over $N = 1000$ samples with a Bernoulli drop-put probabilities of $p = 0.3$ for both caption and image representations. These parameters closely align with Joukovsky et al. (2023). Additionally, we find that scaling the latent representations $\mathbf{z}^a$ and $\mathbf{z}^b$ with the square root of the numbers of tokens $\sqrt{S}$ and image patches $\sqrt{H \times W}$, respectively, helps to stabilize convergence.

Finally, the fitted weight matrix $\mathbf{W}$ models interactions between image patches and caption tokens. Therefore, we can evaluate and visualize it in the same way as our attribution matrices $\mathbf{A}$.

In Section 4.1 we found that InteractionLIME performs good – and even slightly better than our method – on conditional caption attribution. At the same time its conditional image attributions are not competitive. Consequently, its grounding ability as evaluated by the PG-metrics is also weak (cf. Table 2a). We speculate that the reason for this imbalance of attribution quality may be due to the different magnitudes in the number of caption tokens and image patches. While captions typically have ~ 10 tokens, image representations in ViT-B-32 architectures consist of ~ 200 patches. Therefore, the ratio of the number of samples N and tokens is much better than for image patches and the surrogate model φ might be able estimate their importances better.

Overall, we find that the optimization of InteractionLIME is quite sensitive to hyper-parameter choices and requires extensive tuning to find a setting that leads to stable convergence. In contrast, our method does not require additional optimization and involves no hyper-parameters except the number of integration steps N, whose increase must, however, improve attributions due to Equation 9.

H Implementation Details

For the implement of our method, we make use of the auto-differentiation framework in the PyTorch package. For a give input $\mathbf{x}(\alpha_n)$, $\mathbf{g}(\mathbf{x}(\alpha_n))$ is the forward pass through the encoder $\mathbf{g}$, and the Jacobian $\partial \mathbf{g}_k(\mathbf{x}(\alpha_n))/\partial \mathbf{x}_i$ is the corresponding backward pass. For an efficient computation of all N interpolation steps in Eq. 9, we can batch forward and backward passes since individual interpolations are independent of another.
In practice, we attribute to intermediate representations, thus, the interpolations in Eq. 6 are between latent representations of the references and inputs. We use PyTorch *hooks* to compute these interpolations during the forward pass. Algorithm 1 sketches PyTorch-like pseudo-code of the implementation.

The application of our method to a different model or architecture only requires the implementation of a single forward hook. Registering hooks into models is a standard feature in auto-differentiation frameworks and does not require any modification of the given model's original code. The remaining steps to generate our attributions are differentiation through standard backpropagation and, finally, simple matrix multiplication to compute Eq. 10.

I Computational Complexity

Since the computation of the interpolated inputs $\mathbf{x}(\alpha_n)$ can be performed in parallel, N is a constant with regard to time complexity. To build the full Jacobians of the encoders, however, we need to compute a separate backward pass for each output dimension, because auto-differentiation can only compute backward passes for scalar-valued outputs. Time complexity is dominated by this aspect and is thus on the order of $O(D)$, with D being the embedding dimensionality of the output. The `intergrated_jacobian` method in Algorithm 1 sketches this computation.
Due to the fact that we typically attribute to intermediate representations, however, we do not require computation of full backward passes. Backpropagation can be stopped once it reaches the representation we attribute to, which results in this operation to be cheaper the deeper the representation of interest is. I.e. attributing to layer eleven is cheaper than attributing to layer five.
After building the Jacobians, the final attributions are computed through the matrix multiplications in Eq. 10, which is outlined at the bottom of Algorithm 1. Its computation time is negligible when calculated on GPU, but can substantially add to the total time when performed on CPU.

Space-wise, our method requires storing two Jacobians with the dimensions $D \times (D \times S)$, and $D \times (D \times H \times W)$, since input/intermediate representations are still sequential (S) on the text side and patch-based ($H \times W$) on the image side. Thus, memory consumption scales quadratically on the order of $O(D^2)$, and we require large VRAM to handle the computation efficiently on GPU.

In contrast, first-order methods only require backpropagation of a single scalar output value, i.e. the similarity score, whose result is a gradient vector as opposed to a Jacobian matrix. Hence, the cost of obtaining our second-order interaction attribution is the computation and handling of these Jacobians, which is substantially more expensive but enables a different level of insight into models that is not accessible through first-order methods.

Algorithm 1 PyTorch-like pseudocode sketching the computation of our attributions. The syntax is simplified and not logically consistent. For a fully functional implementation, please refer to our GitHub repository. Comments in the pseudocode refer to the corresponding equations in Section 3.

```
from torch import Tensor, arange, stack, autograd
import ExplainableCLIP, image_preparation, tokenize

model         = ExplainableCLIP(...)
image_input   = image_preparation(load_image("path/to/image.png"))
caption_input = tokenize("some caption describing the image")

# Equations 6 and 7
def interpoloate(x: Tensor, ref: Tensor, n_steps: int):
    '''
    Compute n_steps linear interpolations between a reference ref and input x.
    '''
    step = 1 / n_steps
    alphas = arange(1, 0, step)  # interpolation coefficients
    x_interp = ref + alphas * (x - ref)  # interpolated representations
    return x_interp

# Equation 9
def integrated_jacobian(embedding: Tensor, intermediate: Tensor):
    '''
    Compute the integrated Jacobian for an embedding w.r.t. an
    intermediate representation.
    '''
    gradients = []
    for dim in range(embedding.size(0)):
        grad_d = autograd.grad(embedding[dim], intermediate)
        gradients.append(grad_d)
    jacobians = stack(gradients)
    # Integration over interpolations stacked along the first dimension
    int_jacobian = jacobians.sum(dim=0)
    return int_jacobian

# place hooks in the model to compute interpolations (not actual hook syntax)
image_hook   = model.register_hook(interpolate, image_layer, image_reference,   n_steps)
caption_hook = model.register_hook(interpolate, text_layer,  caption_reference, n_steps)

# Compute embeddings and retrieve intermediate representations from hooks
image_embedding   = model.encode_image(image_input)
caption_embedding = model.encode_caption(caption_input)
image_inter, image_ref_inter     = image_hook.get_intermediate_representation()
caption_inter, caption_ref_inter = caption_hook.get_intermediate_representation()

# Equation 10
image_jacobian   = integrated_jacobian(image_embedding,   image_inter)
caption_jacobian = integrated_jacobian(caption_embedding, caption_inter)
# Matrix multiplication
JJ = caption_jacobian.T @ image_jacobian
image_delta   = image_inter   - image_ref_inter
caption_delta = caption_inter - caption_ref_inter
# Element-wise multiplication with broadcasting
attributions = caption_delta * JJ * image_delta
```