A DERIVATIONS OF THE NEGATIVE LOG-LIKELIHOOD

For completeness, we provide a derivation for the negative log-likelihood 2 from Kong et al. (2022). We first introduce a seminal result from Guo et al. (2005),

$$\frac{d}{d\gamma} I(\boldsymbol{x}; \boldsymbol{x}_\alpha) = 1/2 \, \mathrm{mmse}(\gamma). \tag{7}$$

This relationship admits a point-wise generalization,

$$\frac{d}{d\gamma} D_{KL}[p(\boldsymbol{x}_\alpha|\boldsymbol{x}) \parallel p(\boldsymbol{x}_\alpha)] = 1/2 \, \mathrm{mmse}(\boldsymbol{x}, \gamma), \tag{8}$$

The marginal is $p(\boldsymbol{x}_\alpha) = \int p(\boldsymbol{x}_\alpha|\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}$, and the pointwise MMSE is defined as follows,

$$\mathrm{mmse}(\boldsymbol{x}, \gamma) \equiv \mathbb{E}_{p(\boldsymbol{x}_\alpha|\boldsymbol{x})} \left[\|\boldsymbol{x} - \hat{\boldsymbol{x}}^*(\boldsymbol{x}_\alpha, \gamma)\|^2 \right]. \tag{9}$$

To obtain the desired result, we apply the thermodynamic integration trick introduced in Kingma et al. (2021), by first defining the point-wise gap function $f(\boldsymbol{x}, \gamma)$ as

$$f(\boldsymbol{x}, \gamma) \equiv D_{KL}[p(\boldsymbol{x}_\alpha|\boldsymbol{x}) \parallel p_G(\boldsymbol{x}_\alpha)] - D_{KL}[p(\boldsymbol{x}_\alpha|\boldsymbol{x}) \parallel p(\boldsymbol{x}_\alpha)].$$

We denote $p_G(\boldsymbol{x}_\alpha) = \int p(\boldsymbol{x}_\alpha|\boldsymbol{x}) p_G(\boldsymbol{x}) d\boldsymbol{x}$ as the marginal output distribution of the MMSE for the channel with Gaussian input as $\mathrm{mmse}_G(\gamma)$. In the limit of zero SNR, we get $\lim_{\gamma \to 0} f(\boldsymbol{x}, \gamma) = 0$. In the high SNR limit, Kong et al. (2022) prove that

$$\lim_{\gamma \to \infty} f(\boldsymbol{x}, \gamma) = \log \frac{p(\boldsymbol{x})}{p_G(\boldsymbol{x})}. \tag{10}$$

Combining this with Eq. 8, we can write the log likelihood *exactly* in terms of the log likelihood of a Gaussian and a one dimensional integral.

$$\begin{aligned} -\log p(\boldsymbol{x}) &= -\log p_G(\boldsymbol{x}) - \int_0^\infty d\gamma \frac{d}{d\gamma} f(\boldsymbol{x}, \gamma) \\ &= -\log p_G(\boldsymbol{x}) - 1/2 \int_0^\infty d\gamma \left(\mathrm{mmse}_G(\boldsymbol{x}, \gamma) - \mathrm{mmse}(\boldsymbol{x}, \gamma) \right) \end{aligned} \tag{11}$$

This expresses density in terms of a Gaussian density and a correction that measures how much better we can denoise the target distribution than we could using the optimal decoder for Gaussian source data. The density can be further simplified by writing out the Gaussian expressions explicitly and simplifying with an identity given in,

$$-\log p(\boldsymbol{x}) = d/2 \log(2\pi e) - 1/2 \int_0^\infty d\gamma \left(\frac{d}{1+\gamma} - \mathrm{mmse}(\boldsymbol{x}, \gamma) \right). \tag{12}$$

Observe that the first term in the integrand does not depend on $\boldsymbol{x}$, which allows us to derive the desired result Eq. 2. We refer readers to Kong et al. (2022) for more detailed derivations.

B DERIVATIONS OF POINTWISE INFORMATION VIA THE ORTHOGONALITY PRINCIPLE

Our goal is to show that the following expression,

$$\mathrm{i}^o(\boldsymbol{x}; \boldsymbol{y}) \equiv 1/2 \int \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[\|\hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}_\alpha) - \hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}_\alpha|\boldsymbol{y})\|^2 \right] d\alpha,$$

is a pointwise information estimator, i.e., that it satisfies the identity,

$$I(X;Y) = \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}[\mathrm{i}^o(\boldsymbol{x}; \boldsymbol{y})].$$

To show this fact, we first recall the definition of our optimal denoiser and optimal conditional denoiser.

$$\hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}) \equiv \arg\min_{\bar{\boldsymbol{\epsilon}}(\cdot)} \mathbb{E}_{p(\boldsymbol{x}),p(\boldsymbol{\epsilon})} \left[\|\boldsymbol{\epsilon} - \bar{\boldsymbol{\epsilon}}(\boldsymbol{x}_\alpha)\|^2 \right]$$

$$\hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}|\boldsymbol{y}) \equiv \arg\min_{\bar{\boldsymbol{\epsilon}}(\cdot)} \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{y}),p(\boldsymbol{\epsilon})} \left[\|\boldsymbol{\epsilon} - \bar{\boldsymbol{\epsilon}}(\boldsymbol{x}_\alpha|\boldsymbol{y})\|^2 \right]$$

For this optimal denoiser, the following expression holds exactly.

$$\begin{aligned} -\log p(\boldsymbol{x}) &= 1/2 \int \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[\|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}_\alpha)\|^2 \right] d\alpha + const \\ -\log p(\boldsymbol{x}|\boldsymbol{y}) &= 1/2 \int \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[\|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}_\alpha|\boldsymbol{y})\|^2 \right] d\alpha + const \end{aligned} \tag{13}$$

Therefore, we have that,

$$\begin{aligned} I(X;Y) &= \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}[\log p(\boldsymbol{x}|\boldsymbol{y}) - \log p(\boldsymbol{x}] \\ &= \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})} \left[1/2 \int \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[\|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}_\alpha)\|^2 - \|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}_\alpha|\boldsymbol{y})\|^2 \right] d\alpha \right] \end{aligned}$$

Rearranging we have the following.

$$\begin{aligned} I(X;Y) = \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})} \Bigg[& \overbrace{1/2 \int \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[\|\hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}_\alpha) - \hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}_\alpha|\boldsymbol{y})\|^2 \right] d\alpha}^{\mathrm{i}^o(\boldsymbol{x};\boldsymbol{y})} \Bigg] \\ & + 2\mathbb{E}_{p(\boldsymbol{y})} \left[1/2 \int \underbrace{\mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{y}),p(\boldsymbol{\epsilon})} \left[(\hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}_\alpha) - \hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}_\alpha|\boldsymbol{y})) \cdot (\hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}_\alpha|\boldsymbol{y}) - \boldsymbol{\epsilon}) \right]}_{\equiv \mathcal{O}} d\alpha \right] \end{aligned}$$

What remains is to show that the term in red is zero, $\mathcal{O} = 0$, and therefore the whole second term is equal to zero. This fact follows from the orthogonality principle (Kay, 1993), which states the slightly more general result that,

$$\forall \boldsymbol{f}, \;\; \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{y})p(\boldsymbol{\epsilon})}[\boldsymbol{f}(\boldsymbol{x}_\alpha, \boldsymbol{y}) \cdot (\hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}_\alpha|\boldsymbol{y}) - \boldsymbol{\epsilon})] = 0.$$

Note that this is stated in a slightly different way, as we have used $\boldsymbol{x}_\alpha \equiv \boldsymbol{x}_\alpha(\boldsymbol{x}, \boldsymbol{\epsilon})$ to write the noisy channel that our MMSE estimator is attempting to use to recover $\boldsymbol{\epsilon}$. The term $(\hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}_\alpha|\boldsymbol{y}) - \boldsymbol{\epsilon})$ is recognized as the error of the MMSE estimator. This error must be orthogonal to any estimator, $\boldsymbol{f}$. If it isn't, then we can use it to build an estimator with lower MSE than $\hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}_\alpha|\boldsymbol{y})$, contradicting our assumption that $\hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}_\alpha|\boldsymbol{y})$ is the MMSE estimator. A similar result to the orthogonality principle can be shown in a more general way using Bregman divergences (Banerjee et al., 2005).

Therefore, we finally have the desired result that $I(X;Y) = \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}[\mathrm{i}^o(\boldsymbol{x};\boldsymbol{y})]$. Note that this pointwise estimator has a slightly different interpretation from the standard one, i^s, as it is not equal to a log-likelihood ratio pointwise, though it is still in expectation. On the other hand, it has several nice properties. It is non-negative, which is convenient for visualizing heatmaps. It is clear that if mutual information is zero, then the optimal denoiser should learn to ignore $\boldsymbol{y}$, so $\hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}_\alpha|\boldsymbol{y}) = \hat{\boldsymbol{\epsilon}}_\alpha(\boldsymbol{x}_\alpha)$ and our information estimate is then zero.

C Additional Results

C.1 Relationship between image-level MI and CMI

On both the COCO100-IT and COCO-WL datasets, we conducted further calculations for image-level MI and CMI, presenting the results in scatterplots in Fig. 4. These quantitative findings align with our pixel-level visual analysis (§3.2). MI and CMI exhibit strong consistency for noun words, with a high Pearson correlation coefficient of 0.89. In most cases, MI values remain higher than CMI, primarily due to MI containing more information from the background context. However, for abstract words, the Pearson coefficient drops to 0.17, and notably, MI is consistently larger than CMI (with most cases being nearly zero), indicating MI's superior capability in capturing information involving abstract words compared to CMI. This signals a high degree of redundancy between abstract words and context (Williams & Beer, 2010).

C.2 Image-level MMSE curves and pixel-level MMSE visualizaiton

We analyze the image-level (Fig. 5) and pixel-level (Fig. 8 and 9) MMSE for 10 cases in COCO100-IT. To fully harness the capabilities of ITD (Kong et al., 2022), we configured the diffusion steps

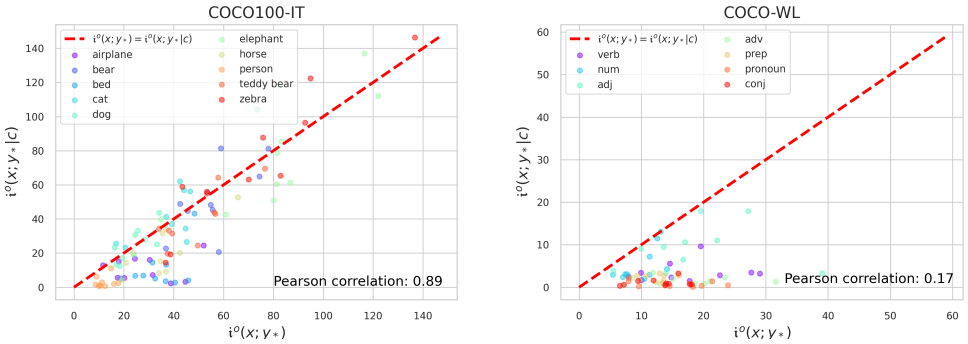

Figure 4: Scatter plot for correlation between MI and CMI.

to be 200. We conducted 50 samples under the same α, and the MMSE results are derived from the average denoising of these samples. However, for the purpose of pixel-level visualization, we selected only 20 steps.

From Fig. 5, it becomes evident that as α varies, the orthogonal approximation exhibits greater stability with fewer zigzag patterns compared to the standard version. Furthermore, the orthogonal method enhances the consistency between MMSE and conditional MMSE, leading to synchronized peaks and similar distributions. The diffusion process reveals that the optimal performance for locating object-related pixels in the image coincides with the appearance of peaks in Fig 5. When the α is too high, the highlighted pixels gradually become sparse, while excessively low α values lead to chaos in MMSE.

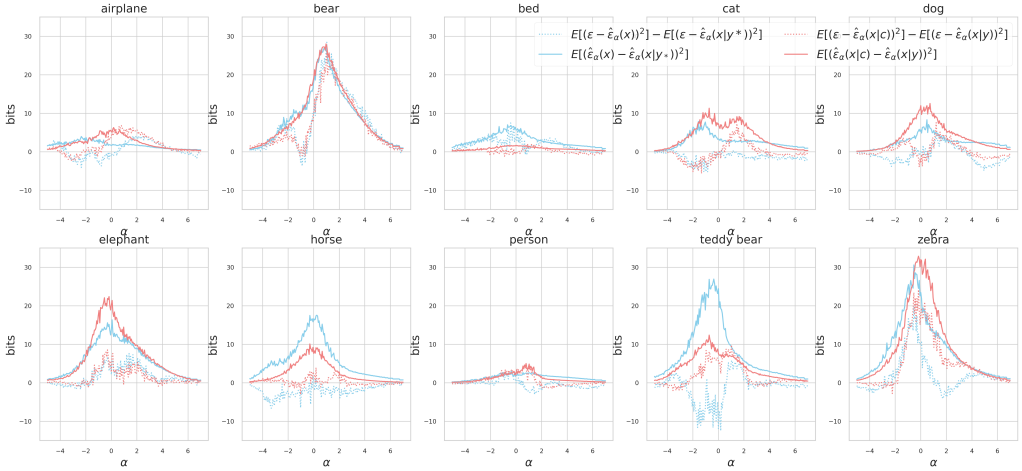

Figure 5: MMSE curves examples for 10 categories.

C.3 Examples of word location for object nouns

We put more pixel-level MI and CMI visualization examples from COCO100-IT, see Fig. 10 & 11 & 12.

C.4 Examples of word location for other seven entities

We put more word location visualization examples for seven entities from COCO-WL, see Fig. 14 & 13.

C.5 INTERVENTION EXPERIMENTS ADDITIONAL RESULTS

(a) Low correlation between image change and heatmaps

(b) High correlation between image change and heatmaps

Figure 6: (a) An example where the correlation between pixel-level changes and CMI or attention are low (0.25 and -0.21 respectively). (b) An example where the pixel-level correlation is high (0.73 and 0.77 respectively).

We observe that correlations between heatmaps (from conditional mutual information or from attention) often correlate strongly with changes in the image after intervention. However, this is not always the case. We show a negative and positive example in Fig. 6. We see in some images that one word change globally changes the image, leading to poor correlation. This result, however, does not contradict our original hypothesis, which is that small CMI implies that omitting a word will have no effect. We generally observe this to be true. However, when the CMI is large, the effect may or may not be correctly localized. The reason that the effect is not always correctly localized is that generation is an iterative procedure: a small change in the first step can lead to global changes in the image.

Fig. 15 and 16 visualize additional examples where we swap a word in a caption with a categorically similar word. For the COCO-IT dataset described in §E, we explored the following word swaps: dog $\leftrightarrow$ cat, zebra $\leftrightarrow$ horse, bed $\leftrightarrow$ table, bear $\leftrightarrow$ elephant, airplane $\leftrightarrow$ kite, person $\leftrightarrow$ clown, plus plural versions. In these plots, and also Fig. 1 and 6, the pixel values represent i^o and are shown with a colormap where the maximum value corresponds to 0.15 bits/pixel. However, the "total information" shown in white text uses the unbiased estimate i^s, and hence can sometimes be negative. Attention color maps are normalized as was done by Tang et al. (2022).

D EXPERIMENTAL SETTINGS

We provide code for reproducing our experiments at `https://github.com/kxh001/Info-Decomp`.

D.1 RELATION TESTING WITH POINTWISE INFORMATION

We refer readers to Table 5 for additional implementation details for evaluating the ARO benchmark. All datasets are prepared following the official implementation of Yuksekgonul et al. (2022) available at `https://github.com/mertyg/vision-language-models-are-bows.git`. All experiments are run on NVIDIA RTX A6000 GPUs.

We evaluate the OpenCLIP checkpoint `laion/CLIP-ViT-H-14-laion2B-s32B-b79K`. This checkpoint consists of a 330M BERT-style encoder trained on the LAION-2B Dataset. Its text encoder is consistent with the one deployed by Stable Diffusion version 2.1 to ensure fair comparison. We use a batch size of 80 for all OpenCLIP evaluations.

Table 5: Additional Experiment Details for the ARO Benchmark

	VG-A	VG-R	COCO	Flickr30k
Perturbation size	1	1	4	4
Dataset size	28,748	23,937	25,010	5,000
Inference batch size	10	10	5	5
SNR sample size	100	100	100	100

In the fourth-row of Table 6, we report the performance of OpenCLIP wherein the last layer of its text encoder is removed. This setup is consistent with Stable Diffusion's usage of text encoder, but we observe the results to be similar for *VG-Relation* and *VG-Attribution* (1 perturbation), and significantly worse for *COCO-Order* and *Flickr30k-Order* (4 perturbations). All other entries are identical to Table 1 for reference.

We report fine-grained performance of Stable Diffusion and OpenCLIP systems across each relation type in Table 10. In column 3, we report normalized prediction disagreement between uniform and logistic sampling, and observe the predictions to be generally consistent.

Additionally, we assess the consistency of our estimator across random seeds. For each dataset, we select the first 1000 samples, evaluate our estimator across 3 random seeds, and provide OpenCLIP baseline on the same subsets as for reference. Numerical results are provided in Table 7. The information estimates are relatively consistent, and establish statistically significant performance gap compared to OpenCLIP.

Table 6: Additional Accuracy (%) of Stable Diffusion and OpenCLIP.

Method	VG-A	VG-R	COCO	Flickr30k
Baseline (Random Guess)	50.0	50.0	20.0	20.0
OpenCLIP Ilharco et al. (2021)	64.6	51.4	32.8	40.5
OpenCLIP (all-but-last)	65.6	50.9	22.4	28.8
Info. (Ours, Uniform)	71.2	68.5	39.3	48.7
Info. (Ours, Logistic)	**72.0**	**69.1**	**40.1**	**49.3**

Table 7: Mean Estimator Accuracy and Std. Dev. across Random Seeds

	VG-A	VG-R	COCO	Flickr30k
	Acc.±Std. Dev. (%)			
Info. (Unif.)	71.7 ± 0.59	72.3 ± 2.87	36.3 ± 0.87	50.5 ± 0.63
Info. (Log.)	$\mathbf{72.7} \pm 1.32$	$\mathbf{73.4} \pm 1.13$	$\mathbf{36.8} \pm 0.50$	$\mathbf{51.2} \pm 0.73$

D.2 Localizing Word Information in Images

In our word localization experiments, we utilized a pre-trained Stable Diffusion v2-1-base model card available at Huggingface. Input images were resized to 512×512 and then normalized to the [0, 1] pixel value range to ensure compatibility with the pre-trained model.

The DAAM Tang et al. (2022) is essentially an extension integrated into Stable Diffusion models, designed to generate attention-based heatmaps concurrently with the image generation process. To leverage DAAM, it is imperative to pair it with a diffusion scheduler. In our experiments, we draw inspiration from Liu et al. (2023) and employ a DDIM Song et al. (2022) scheduler as a baseline. While our ITD model Kong et al. (2022) is capable of independently generating MI and CMI heatmaps using the principles outlined in §2, we also had the option to enhance attention heatmaps by integrating DAAM with the information-theoretic diffusion process. Hence, we established three sets of comparative experiments: DAAM+DDIM (Attention), ITD (CMI), and DAAM+ITD (Attention+Info.) respectively. We opt not to use classifier-free guidance since it primarily aids in image generation and introduces additional undesired content onto images in the denoising process. We

utilize the "alphas_cumprod" in the scheduler of the Stable Diffusion model to compute the α range spans from -5 to 7. For specific t-to-α transformation calculations, you could refer to Appendix B.2 in Kong et al. (2022). Thus, the parameters of the corresponding logistic distribution are [loc, scale, clip] = [1, 2, 3]. Unfortunately, DAAM only supports a batch size of 1. For DAAM+DDIM and DAAM+ITD, we utilize the following input: ($\boldsymbol{x}$, $\boldsymbol{y}$, $y*$). In the case of ITD, the input configuration is ($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{c}$). This distinction arises from the fact that DAAM generates heatmaps based on the cross-attention map, enabling the direct calculation of the score for an individual object word on each pixel. On the other hand, ITD relies on conditional mutual information, $\mathrm{i}_j^o(\boldsymbol{x}; y_*|\boldsymbol{c})$.

Table 8: The hyper-parameters used in DDIM and ITD schedulers.

random seed	batch size (DDIM/ITD)	logistic distribution (ITD)	guidance (DDIM)
42	1/10	[loc, scale, clip] = [1, 2, 3]	1

All experiments were conducted using Nvidia RTX 6000 GPU cards. The hyper-parameters used in these experiments are summarized in Table 8, with variations in the number of diffusion steps set at 50, 100, and 200. Once the heatmap of the image is computed, we initially rescale it to the [0, 1] range and subsequently apply a uniform hard threshold on them for segmentation. After experimenting with hard thresholds vary in [0, 1], we identify the optimal threshold that yields the highest mIoU value, then record the mIoU in Table 3. Unless explicitly stated, all visualization for MI, CMI, and attention heatmaps are based on 100 diffusion steps.

Table 9: Unsupervised Object Segmentation mIoU (%) Standard Error Analysis on COCO-IT

Method	1 step	50 steps	100 steps	200 steps
Whole Image Mask	14.94 ± 0.0022	14.94 ± 0.0022	14.94 ± 0.0022	14.94 ± 0.0022
Attention	37.89 ± 0.0030	34.52 ± 0.0030	34.90 ± 0.0030	35.35 ± 0.0030
CMI	21.73 ± 0.0023	32.31 ± 0.0026	33.24 ± 0.0026	33.63 ± 0.0026
Attention+Info.	$\mathbf{37.96} \pm 0.0030$	$\mathbf{42.46} \pm 0.0032$	$\mathbf{42.71} \pm 0.0032$	$\mathbf{42.84} \pm 0.0032$

We calculated the standard error of the IoU values for object segmentation experiments conducted on COCO-IT, and the results are documented in Table 9. This indicates that the number of diffusion steps does not significantly affect the variation in IoU values. Notice that Table 9 includes an additional column for the 1-step experiment results. The 1-step DAAM-DDIM diffusion process can be regarded as denoising images with imperceptible noise, which is surprisingly effective compared to the multi-step results. However, computing MI and CMI only at a single step, or α, is not directly comparable. The steps in that case are interpreted as elements in a sum approximating an integral, and we don't expect a one step sum to be a good estimate. Additionally, as per the analysis in §C.2, peaks are required for an accurate match between relevant pixels and object words, which cannot be predicted in advance. Nonetheless, the results still demonstrate that the information-theoretic diffusion process enhances attention with respect to object segmentation. Additionally, it's noteworthy to mention that the generation process for MI or CMI from ITD differs from the generation of DAAM. DAAM requires continuous noise addition and denoising iterations to compute, while ITD first samples a series of α, and then each α can undergo independent noise addition and denoising computations. Finally, MI or CMI is calculated by one integration, which facilitates parallel computing, see Fig. 7.

E COCO-IT Dataset Preparation

While the MSCOCO Lin et al. (2015) dataset boasts ample image-text pairs, not every object present in the images is mentioned in the captions, even if these objects have been labeled and annotated. In our experiments, we would like to test (1) the mutual information between a complete prompt and the corresponding image, and (2) the conditional mutual information between the object word and the image. Therefore, we filtered the original COCO 2017 validation dataset using the following steps:

(a) Traverse all the objects in an image.

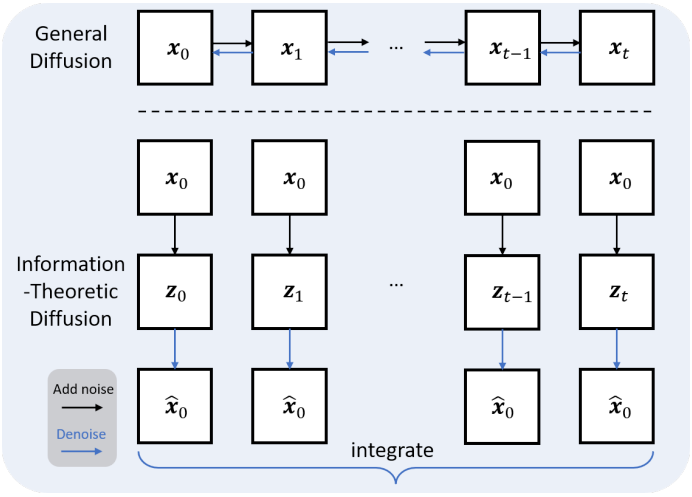

Figure 7: The diagram of two different diffusion processes.

(b) Match each object word to the caption containing that object.

(c) Generate one data point with four contains: [image, caption, context, object].

(d) If one object doesn't appear in the captions, then omit that data point.

After applying this filter, we acquired a dataset, COCO-IT, comprising 6,927 validation image-text data points and 79 categories. To facilitate more effective visualization, we further randomly selected 10 categories from it, choosing 10 image-text pairs for each to create a smaller dataset, COCO100-IT. Additionally, we constructed a dataset, COCO-WL, for word localization by selecting 10 cases for seven different entities (verb, num., adj., adv., prep., pron., conj.).

Table 10: Fine-grained results in Visual Genome Relation dataset.

	Info. (Unif.) (↑)	Info. (Log.) (↑)	Disagreement (↓)	OpenCLIP (↑)	# Samples
Accuracy (%)	68.5	69.1	6.7	51.4	
Spatial Relationships					
above	49.8	53.2	5.6	55.0	269
at	70.7	72.0	9.3	66.7	75
behind	39.9	39.9	4.5	54.4	574
below	49.3	46.4	7.7	49.8	209
beneath	50.0	50.0	0.0	90.0	10
in	76.7	79.9	5.5	51.6	708
in front of	70.2	68.7	7.3	63.1	588
inside	69.0	74.1	8.6	56.9	58
on	75.1	75.6	6.4	51.0	1684
on top of	62.7	63.7	9.0	46.3	201
to the left of	51.1	51.2	7.8	50.5	7741
to the right of	48.6	49.3	7.9	49.8	7741
under	47.0	46.2	3.8	43.9	132
Verbs					
carrying	58.3	66.7	8.3	33.3	12
covered by	36.1	33.3	8.3	55.6	36
covered in	14.3	14.3	14.3	50.0	14
covered with	18.8	18.8	0.0	43.8	16
covering	63.6	72.7	15.2	54.5	33
cutting	91.7	91.7	0.0	66.7	12
eating	85.7	85.7	0.0	57.1	21
feeding	40.0	50.0	10.0	100.0	10
grazing on	60.0	60.0	0.0	30.0	10
hanging on	57.1	71.4	14.3	78.6	14
holding	90.1	87.3	5.6	52.1	142
leaning on	66.7	66.7	0.0	66.7	12
looking at	80.6	83.9	3.2	48.4	31
lying in	100.0	100.0	0.0	33.3	15
lying on	81.7	86.7	5.0	40.0	60
parked on	76.2	71.4	4.8	61.9	21
reflected in	71.4	64.3	7.1	61.9	14
resting on	69.2	84.6	15.4	15.4	13
riding	80.4	76.5	7.8	37.3	51
sitting at	65.4	69.2	3.8	38.5	26
sitting in	82.6	82.6	8.7	65.2	23
sitting on	80.6	78.9	8.6	49.7	175
sitting on top of	80.0	50.0	30.0	60.0	10
standing by	91.7	91.7	16.7	50.0	12
standing in	89.8	91.5	8.5	49.2	59
standing on	78.8	82.7	3.8	55.8	52
surrounded by	64.3	57.1	7.1	42.9	14
using	100.0	100.0	0.0	21.1	19
walking in	90.0	90.0	0.0	70.0	10
walking on	94.7	94.7	0.0	36.8	19
watching	72.7	77.3	4.5	31.8	22
wearing	82.7	84.1	6.4	44.9	949

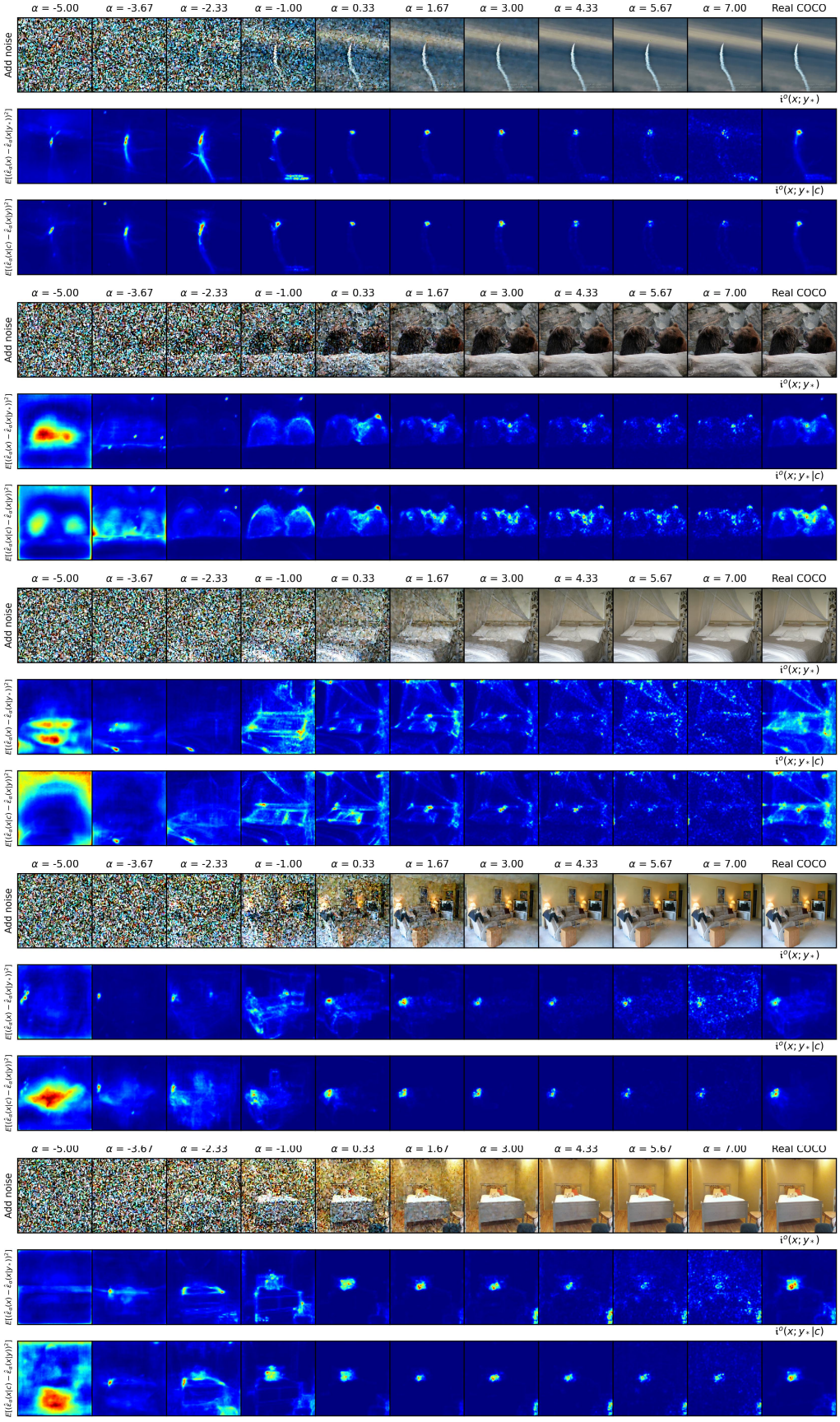

Figure 8: Examples of pixel-level MMSE visualization.

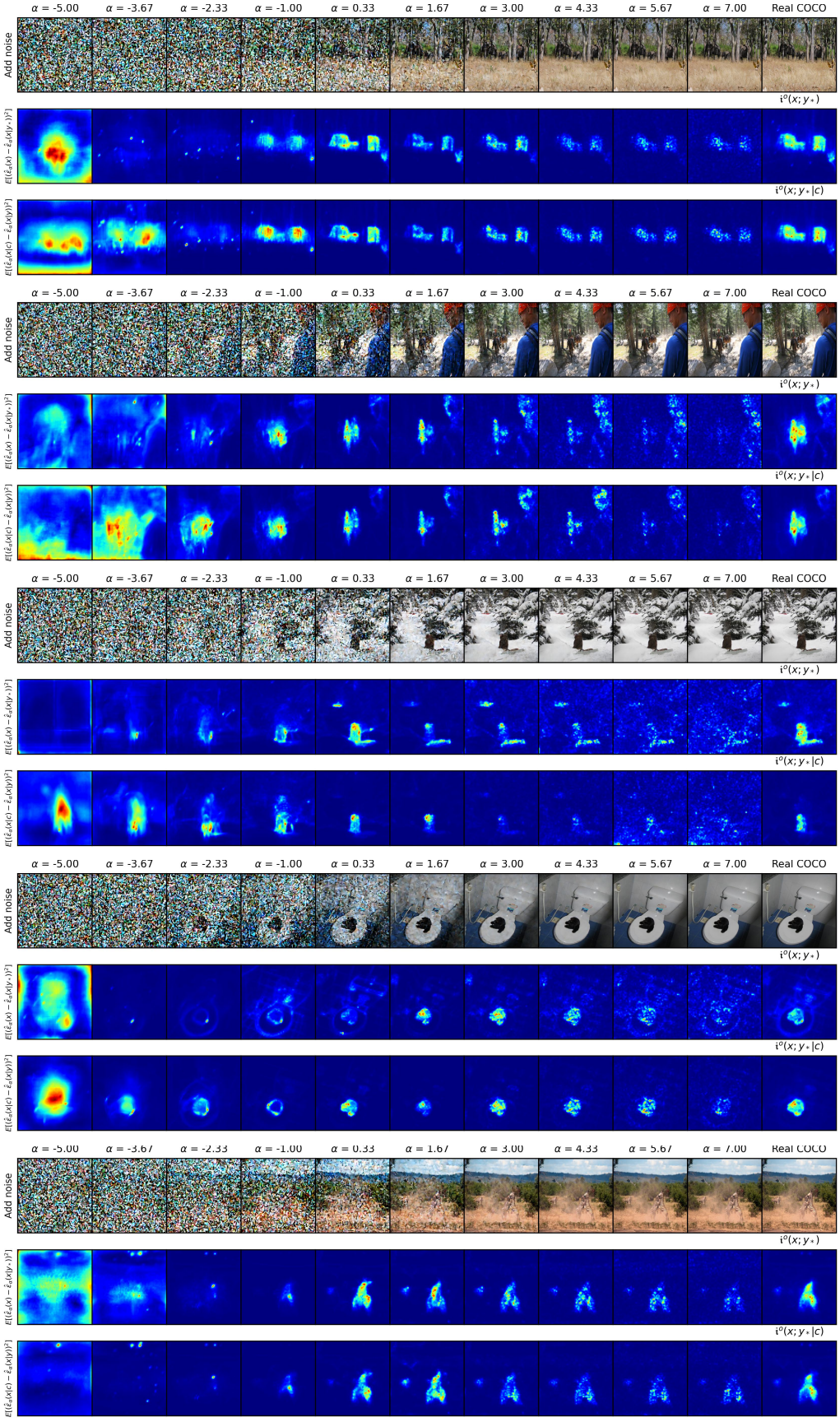

Figure 9: Examples of pixel-level MMSE visualization.

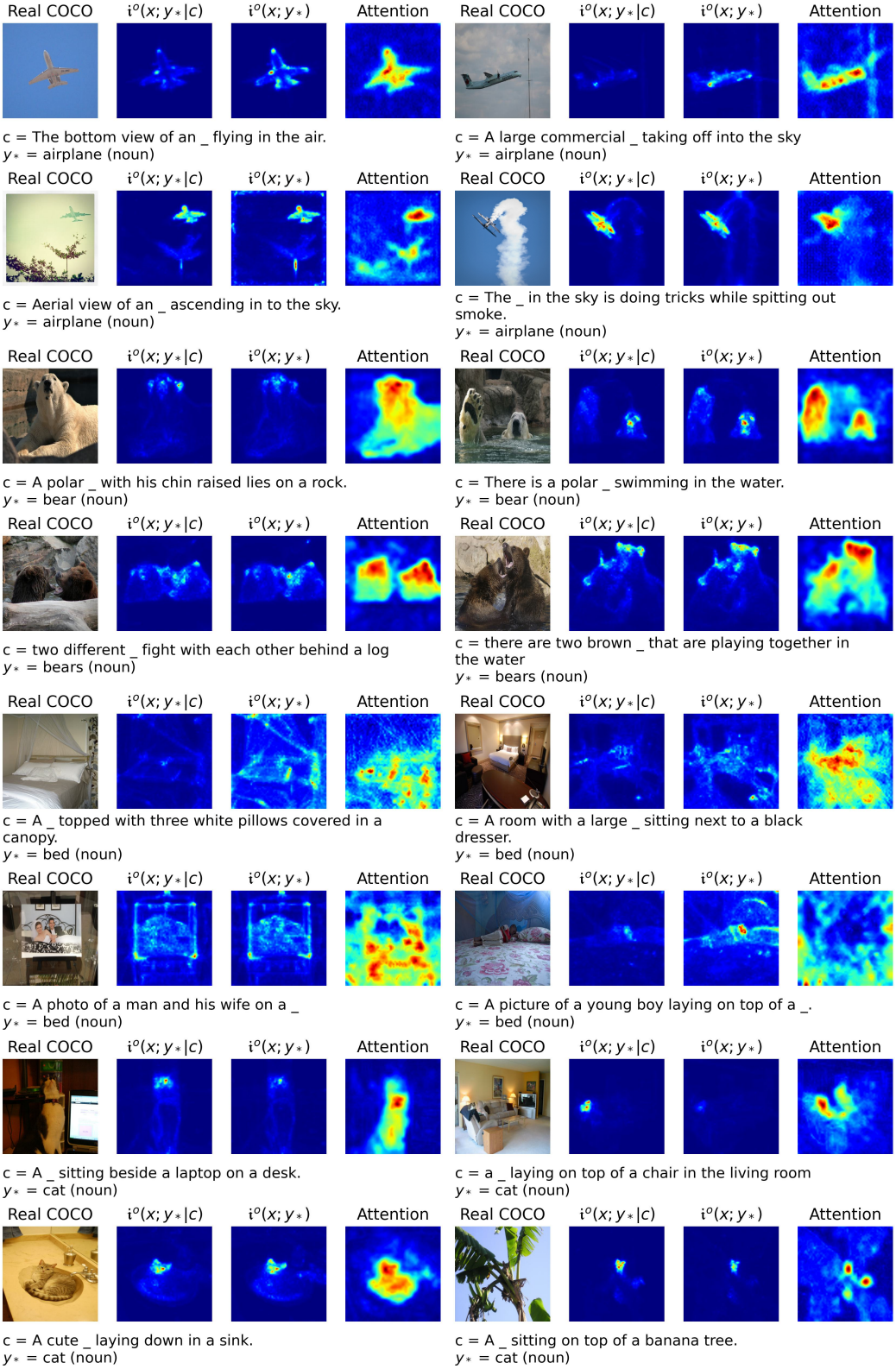

Figure 10: Examples of localizing noun words in images.

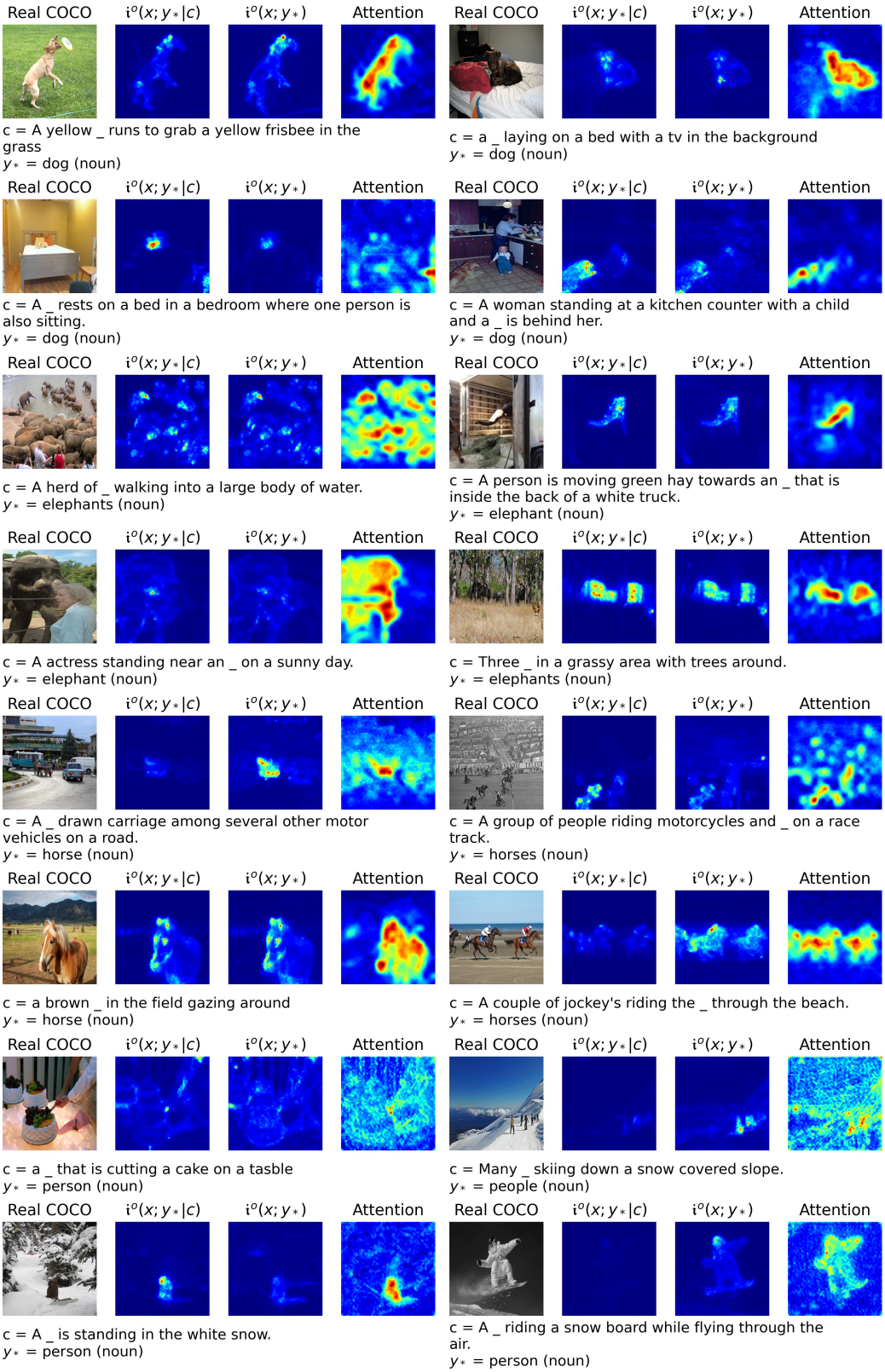

Figure 11: Examples of localizing noun words in images.

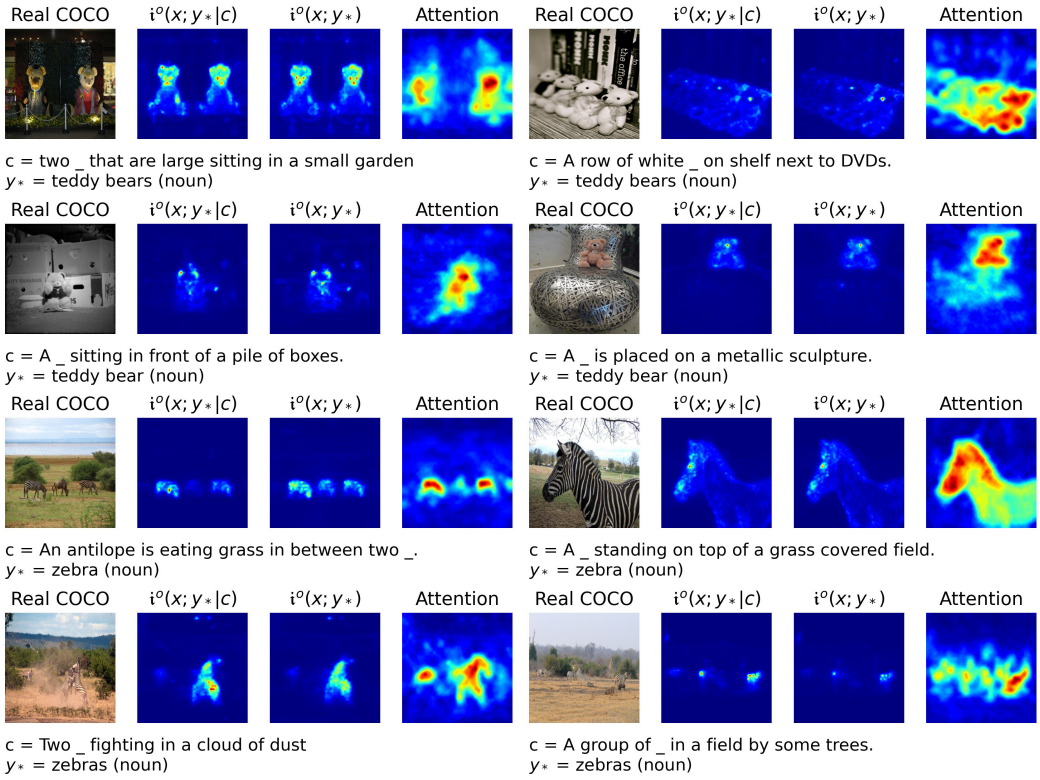

Figure 12: Examples of localizing noun words in images.

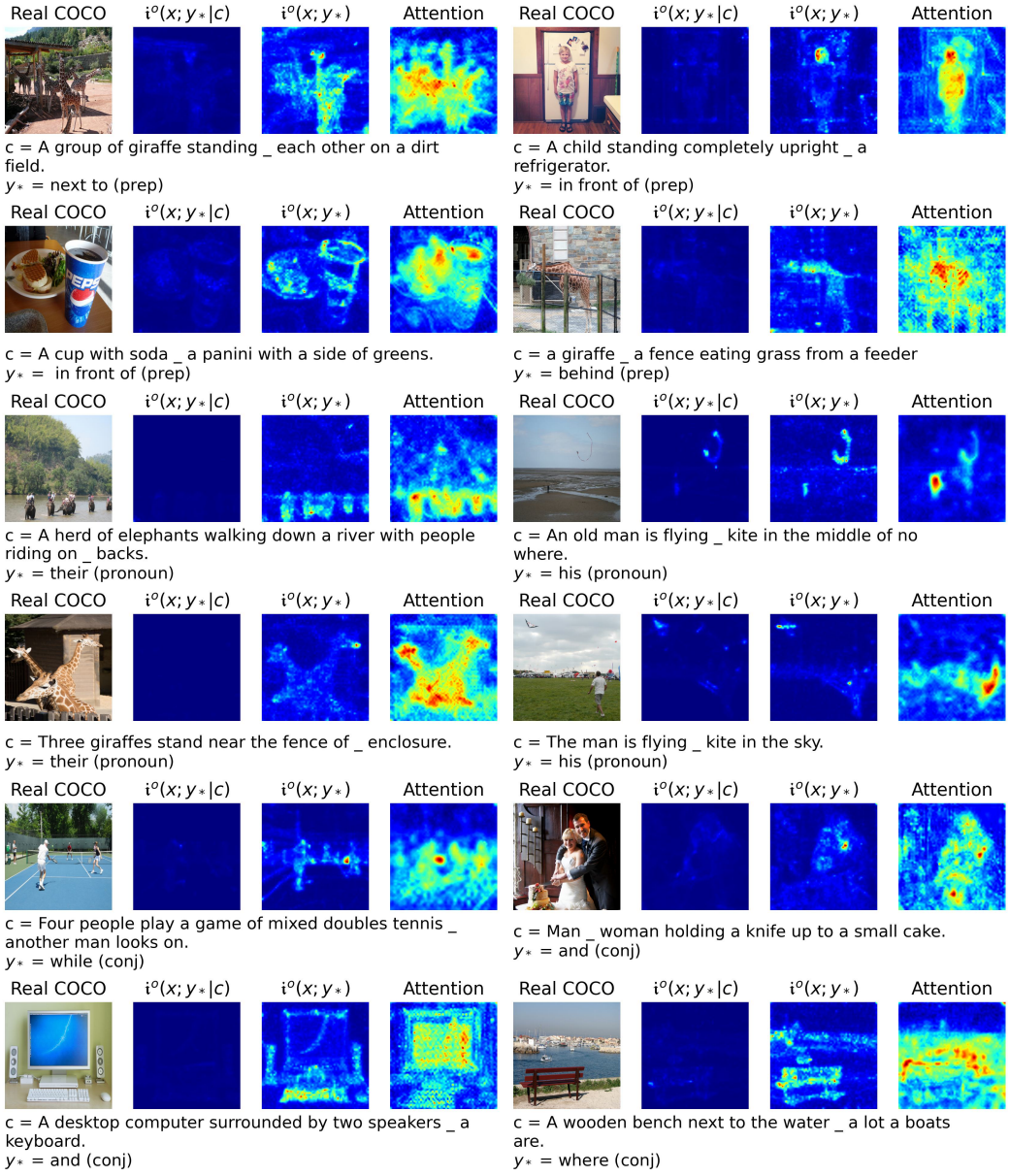

Figure 13: Examples of localizing abstract words in images.

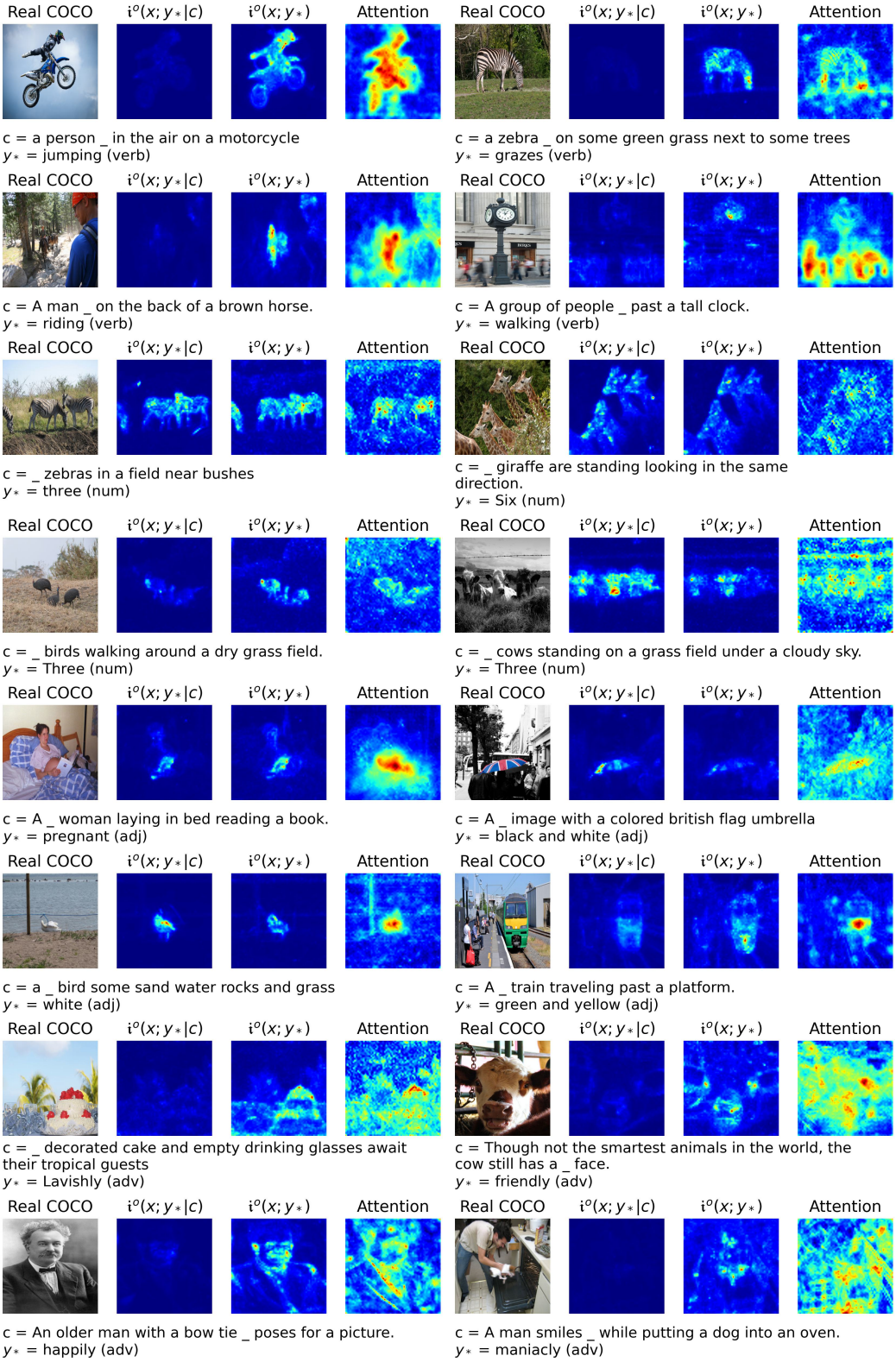

Figure 14: Examples of localizing abstract words in images.

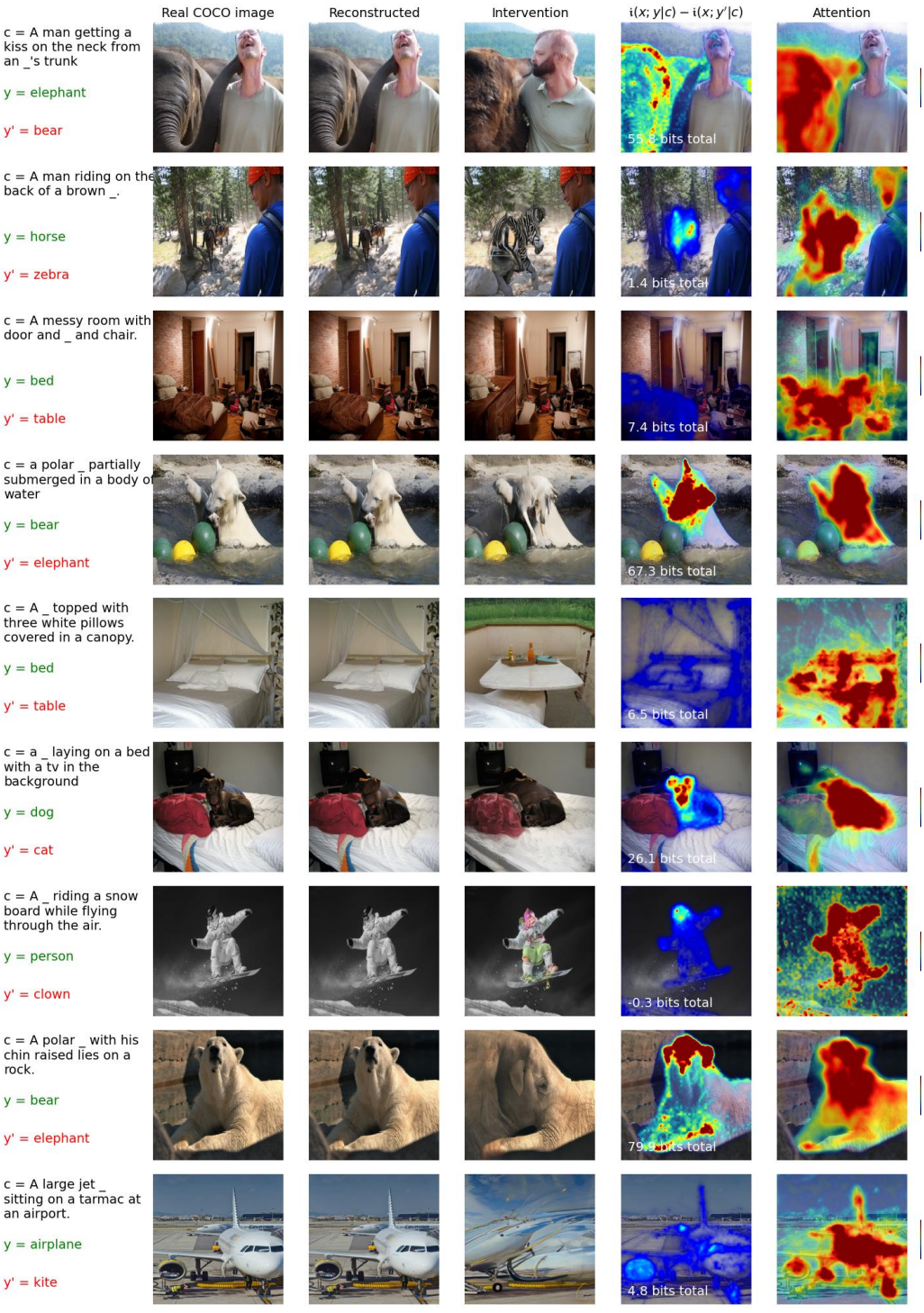

Figure 15: Examples of word swap interventions

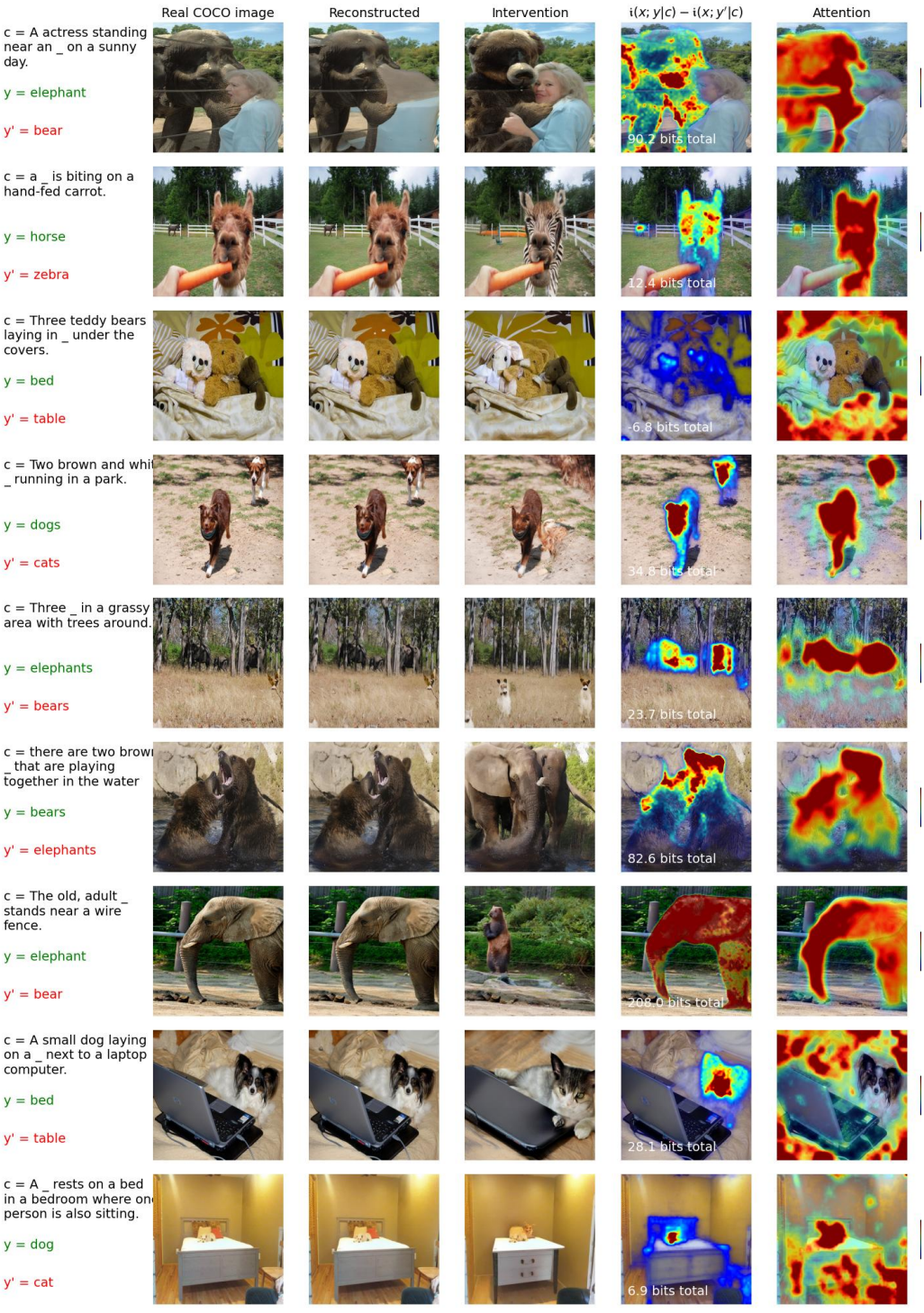

Figure 16: Examples of word swap interventions

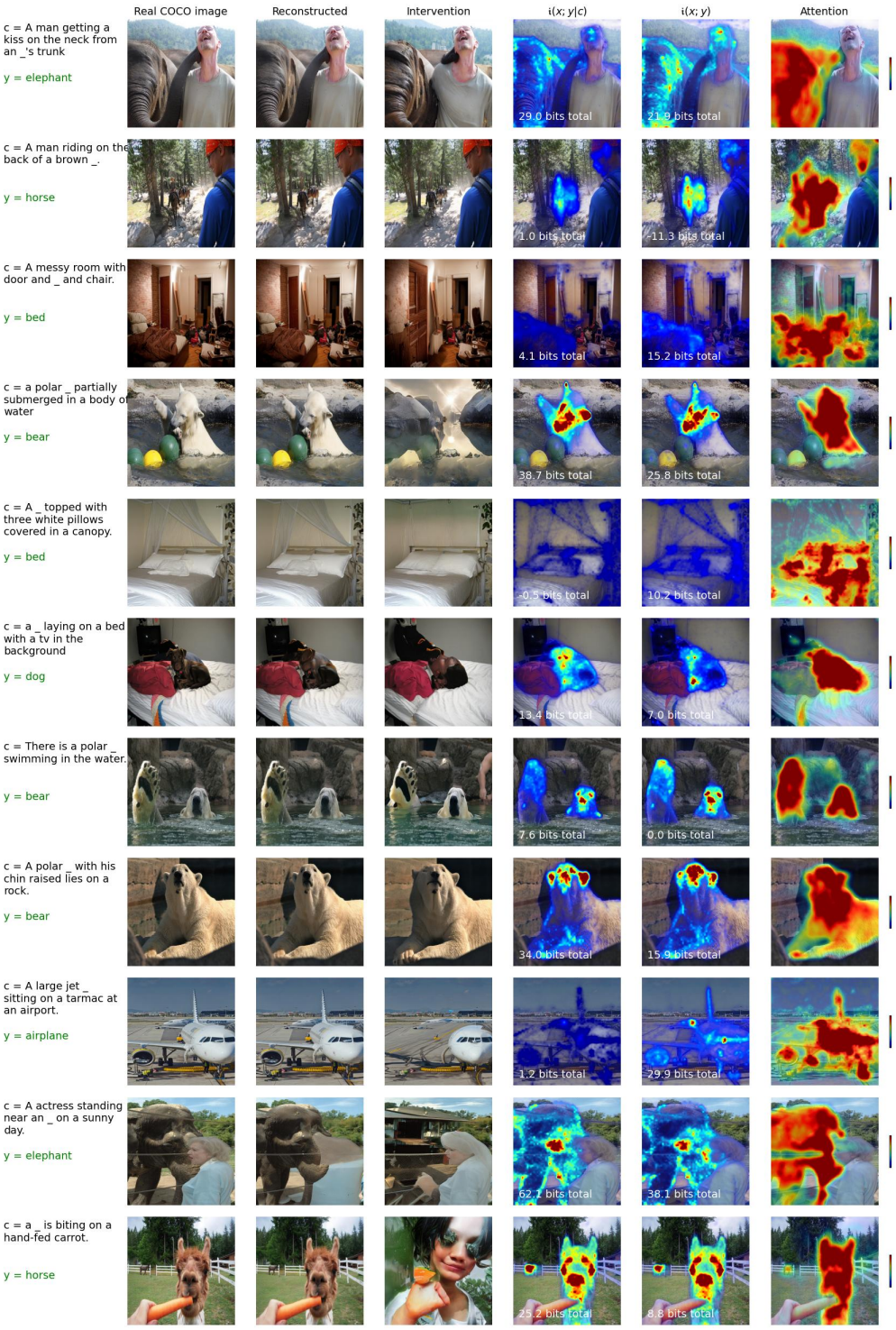

Figure 17: Examples of word omission interventions

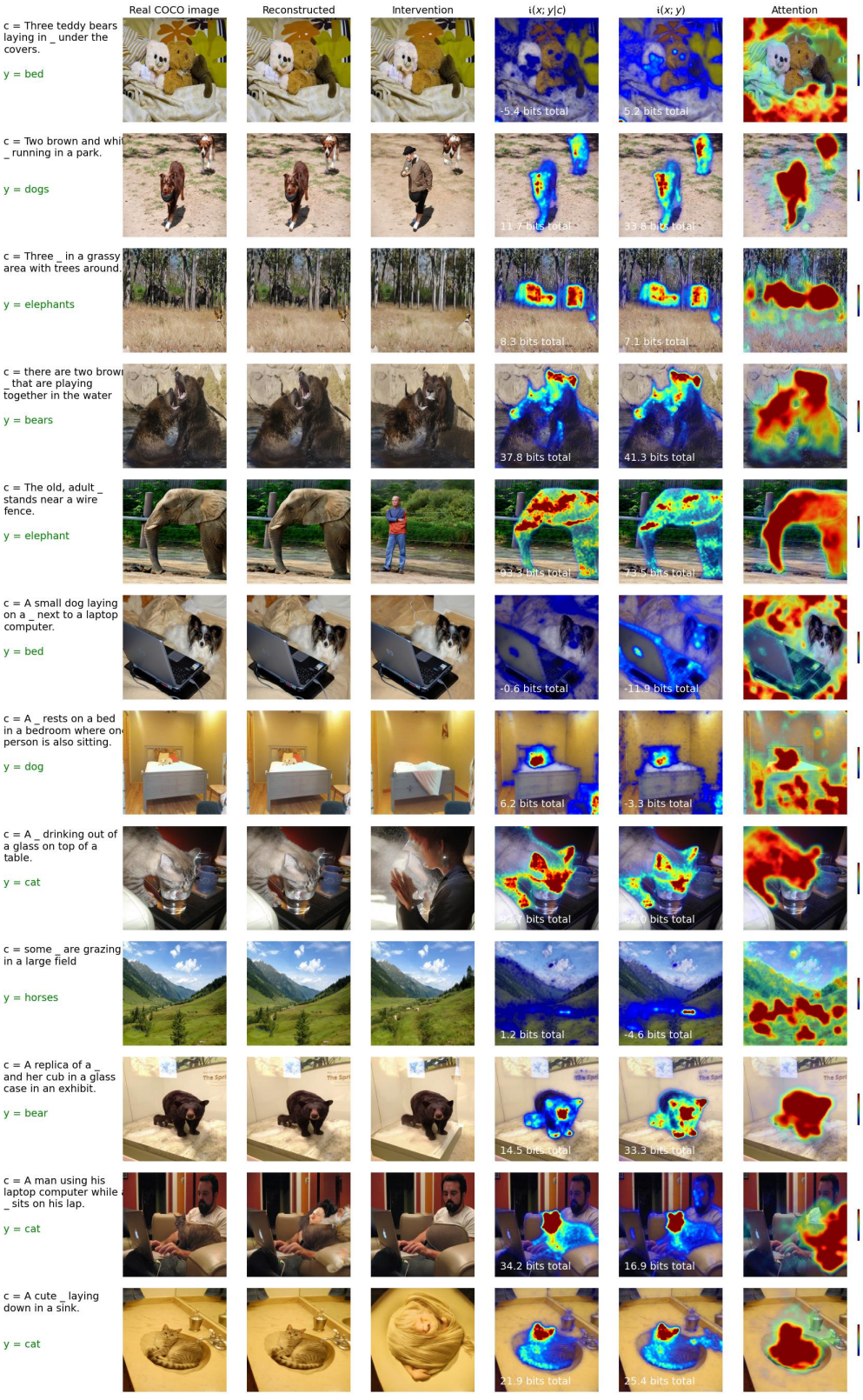

Figure 18: Examples of word omission interventions