No Token Left Behind: Explainability-Aided Image Classification and Generation

Roni Paiss[1], Hila Chefer, and Lior Wolf

The Blavatnik School of Computer Science, Tel Aviv University

Abstract. The application of zero-shot learning in computer vision has been revolutionized by the use of image-text matching models. The most notable example, CLIP, has been widely used for both zero-shot classification and guiding generative models with a text prompt. However, the zero-shot use of CLIP is unstable with respect to the phrasing of the input text, making it necessary to carefully engineer the prompts used. We find that this instability stems from a selective similarity score, which is based only on a subset of the semantically meaningful input tokens. To mitigate it, we present a novel explainability-based approach, which adds a loss term to ensure that CLIP focuses on all relevant semantic parts of the input, in addition to employing the CLIP similarity loss used in previous works. When applied to one-shot classification through prompt engineering, our method yields an improvement in the recognition rate, without additional training or fine-tuning. Additionally, we show that CLIP guidance of generative models using our method significantly improves the generated images. Finally, we demonstrate a novel use of CLIP guidance for text-based image generation with spatial conditioning on object location, by requiring the image explainability heatmap for each object to be confined to a pre-determined bounding box. Our code is available at https://github.com/apple/ml-no-token-left-behind.

1 Introduction

State-of-the-art computer vision models are often trained as task-specific models that infer a fixed number of labels. In contrast, [29] have demonstrated that by training an image-text matching model that employs Transformers for encoding each modality, tens of downstream tasks can be performed without further training ("zero-shot"), with comparable accuracy to the state of the art [29]. Due to its zero shot capabilities and its semantic latent space, CLIP [29] has been widely used in recent research to guide pretrained generative networks to create images according to a text prompt [15, 10, 23] and edit an input image according to a text description [20, 28, 26].

While CLIP shows great promise in zero-shot image classification tasks and generative network guidance, it suffers from instabilities that often lead to biased similarity scores between non-matching text and image pairs. To mitigate

[1] Work was done while the author was also working at Apple.

these instabilities, [29] suggest a prompt engineering technique that averages the embeddings of multiple textual templates. Since CLIP summarizes the relation between a given image and a given text with a single similarity score, it can present a myopic behavior and focus on specific elements within the sentence and/or the image. In order to alleviate this issue, it is necessary to rely on an additional signal. In this work, we propose using the explainability maps to steer the optimization process towards solutions that rely on the relevant parts of the input, and away from solutions that focus on irrelevant parts, or on a small subset of the relevant parts. We explore two domains in which we guide CLIP to account for the important tokens in an input prompt: one-shot classification and zero-shot text-based image generation. For one-shot classification, we incorporate a loss based on the explainability scores of the class name in the prompt engineering method proposed by [42]. Our results demonstrate that guiding the prompt engineering process using explainability improves performance in both the class-agnostic and the class-specific cases. In the domain of image editing guided by text, we employ a similar explainability-based loss. This loss allows the generative network to avoid local minima caused by focusing on irrelevant words in the input text, by requiring the explainability scores of important tokens to be high. We demonstrate that our method significantly improves the generated images. Additionally, by applying similar principles to the image-side relevancy map, we use the obtained heatmaps to facilitate CLIP-guided text-based image generation with spatial conditioning. As far as we can ascertain, we are the first to present a spatial layout to image method using CLIP. As we demonstrate, a straightforward application of the similarity score over a requested bounding box does not guarantee that the entire object will be contained within that bounding box. When relying on the explainability heatmap, our method helps ensure that the object does not deviate from the provided bounding box.

2 Related Work

Zero-shot classification Zero-shot classification in computer vision usually refers to a model's ability to generalize to unseen labels. While several works used weakly labeled Google images as training data [3, 9, 14, 32, 35, 38, 40] the method of [21] was perhaps the first to study zero-shot transfer learning to unseen datasets, which is a broader approach to zero-shot classification. This approach was adopted by CLIP [29], which trains an image-text matching engine using an image encoder and a text encoder, via contrastive learning. The image encoder architectures used are either ViT [12] or ResNet [16], and the text encoder is based on a Transformer [37] with the modifications of [30]. CLIP was trained on a set of $400M$ matching image-text pairs, and showed remarkable zero-shot capabilities on the ImageNet dataset [11]. Following CLIP, [42] proposed few-shot prompt engineering to enhance CLIP's classification accuracy on unseen datasets. Their approach opts to learn the textual templates fed to CLIP rather than manually engineering them, as originally done by [29]. As we show, CLIP-guided optimization methods such as CoOp [42] tend to focus on a sparse set of

tokens in the input, and often neglect important parts of it. Our work attempts to mitigate this issue by applying an explainability-based loss.

CLIP-guided Generation Following the success of CLIP in zero-shot classification, several works have used the similarity scores produced by CLIP to guide pretrained generative networks. These methods usually construct a similarity-based loss, encouraging a generator to produce the output with the desired semantic attributes. Some of the applications of CLIP for guiding generative networks include image manipulation [28], image essence transfer [6], style transfer [15, 43], 3D object style editing [25], and image captioning [36]. While demonstrating great capabilities, we show that methods such as StyleCLIP and VQ-GAN+CLIP are limited by the tendency of CLIP to sometimes ignore meaningful parts of the text. Our explainability-based loss addresses this issue.

Transformer Explainability Many methods were suggested for generating a heatmap that indicates local relevancy, given an input image and a CNN [33, 4, 24, 34]. However, the literature on Transformer explainability is relatively sparse and most methods focus on pure self-attention architectures [1, 8]. The recent method of [7], which we employ in this work, is the first to also offer a comprehensive method for bi-modal networks.

3 Method

We begin by describing how to produce relevance values for each word and image patch using CLIP. We then describe how our method is applied to one-shot classification via prompt engineering and to zero-shot image generation.

3.1 Explainability

Given a pair of text t and image i, CLIP produces a score $\text{CLIP}(t, i)$, which determines how semantically similar the textual description t is to the image i. The goal of the explainability method is to produce a relevance score for each input text token and image patch in the computation of the similarity score $\text{CLIP}(t, i)$. The score of each token should reflect its impact on the similarity score, and input tokens with the greatest influence on the similarity score should receive a high relevancy score and vice versa. We employ the method described in [7] (details in the supplementary) to produce a relevance score for each image patch and text token, given the calculated similarity score. As the relevance scores are calculated per token, we define the relevance score of a word to be the maximal relevance score among its tokens. For each word $W = w_1, ..., w_n$, where $w_1, ...w_n$ are the tokens it contains, we define its relevance score $\mathcal{R}_{expl}(W)$ as $\mathcal{R}_{expl}(W) = \max_{k \in w_1,...,w_n} \mathbf{R}[k]$, where $\mathbf{R}[k]$ is the relevance score of [7].

3.2 Prompt engineering

Image classification is the task of assigning a label from a set of possible classes $c \in C$ to an input image i. In order to adapt CLIP to different classification

tasks, [29] propose employing prompt templates with each possible class $c \in C$ inserted, e.g. "A photo of a {label}." These templates are necessary because in the process of CLIP's pre-training most textual inputs are full sentences rather than single words. Let i be the input image to be classified, and let t denote the textual template. $t(c)$ denotes the template t, where the {label} placeholder was replaced by the class name c. CLIP scores per class are obtained using the similarity score between the input image and the class-template as follows:

$$\Pr(\text{output} = c|i) = \frac{e^{\text{CLIP}(t(c),i)}}{\sum_{c' \in C} e^{\text{CLIP}(t(c'),i)}}. \tag{1}$$

[42] replace the manual selection of the textual templates with a few-shot learning of it. Given the desired template length M, the template $t(\text{label}) = v_1, ..., v_i, \{\text{label}\}, v_{i+1}, ..., v_M$ is optimized with a cross-entropy loss using Eq. 1 to extract the distribution. Note that the learned templates are prone to overfitting, due to the small number of example images for each label, which can result in prompts that describe distinctive parts of the images that are irrelevant to their class, yielding a similarity score $\text{CLIP}(t(c), i)$ that is not based on the class name. This problem is most prominent in the one-shot scenario where the prompts are optimized based on a single image per class. To help avoid this phenomenon, our method employs a novel explainability-based loss. For each class $c \in C$ and image i, the similarity score $\text{CLIP}(t(c), i)$ is produced, and then a normalized explainability score is calculated. This score reflects the relevance of the class $c \in C$ to the similarity of the template and the image:

$$\mathcal{S}_{expl}(c) = \frac{\max_{W \in c} \mathcal{R}_{expl}(W)}{\sum_{U \in t(c)/c} \mathcal{R}_{expl}(U)} \tag{2}$$

where, as above, $W \in c$ are the words comprising the label $c \in C$. The score $\mathcal{S}_{expl}(c)$ encapsulates the impact that the class name c has on the calculated similarity score $\text{CLIP}(t(c), i)$, in comparison to all other words in the sentence. Our explainability-based loss is, therefore:

$$\mathcal{L}_{expl} = \lambda_{expl} \left(-\mathcal{S}_{expl}(c_{gt}) + \sum_{c \neq c_{gt} \in C} \mathcal{S}_{expl}(c) \right)$$

where c_{gt} is the ground truth label, and λ_{expl} is a hyperparameter. The first term, $-\mathcal{S}_{expl}(c_{gt})$ encourages the similarity score for the ground truth class to be based on the class name tokens, in order to avoid focusing on other, irrelevant tokens. The second term, $\sum_{c \neq c_{gt} \in C} \mathcal{S}_{expl}(c)$ encourages the similarity score for all the counterfactual classes to be based on tokens that are not the false class name, since the class name does not correspond to the image.

3.3 Zero-shot text-guided image manipulation

Recent research [28, 20] demonstrates that CLIP can be effective for guiding generative networks that synthesize and edit images, by maximizing the similarity

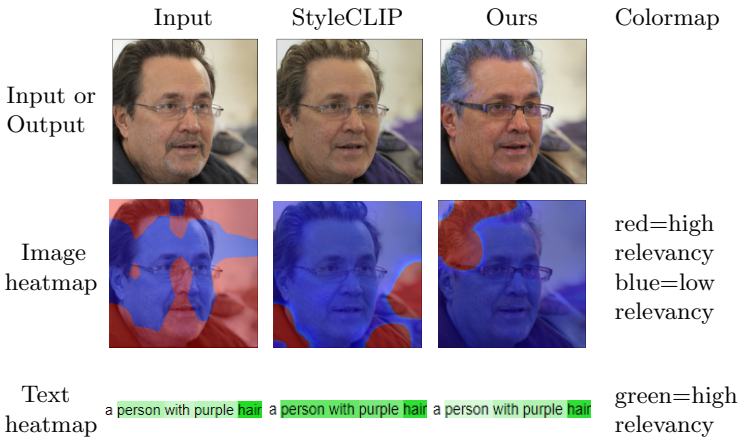

Fig. 1. Manipulations for "A person with purple hair". StyleClip [28] produces a manipulation that is not consistent with the semantic meaning of the prompt, and the color of the person's shirt and the background are altered. Our method generates an output that is faithful to the input text query, and the high values of the explainability heatmaps are much more correlated with the prompt.

score between the desired text and image. As pointed out by [23], methods that integrate a pre-trained generator with the CLIP score to allow text-based editing suffer from instabilities, since similarity-based optimization often reaches local minima that do not reflect the desired semantic meaning of the input query.

As shown in Fig. 1, this shortcoming is often manifested in the explainability scores, and is caused by similarity scores relying only on a partial subset of the semantic tokens in the text query. Thus, our method leverages the explainability scores, to ensure that the output similarity scores are derived from all of the tokens that determine the semantic meaning of the input text. Given a pre-trained generator G (a mapping from a latent vector z to an output image), an input image i, and an input text t, the goal of the optimization algorithm A is to find a vector $A(G, i, t) = z$ such that $G(z)$ combines the visual appearance of image i with the description in the text query t. In order to assess the correlation between the manipulation $G(z)$ and the desired semantic property in t, the algorithm A uses the CLIP similarity score between the manipulated image $G(z)$ and the textual prompt t as a loss term $\mathcal{L}_{similarity} = -\text{CLIP}(G(z), t)$. This loss is applied in addition to other loss terms that lead to a high visual similarity between i and $G(z)$.

As mentioned, $\mathcal{L}_{similarity}$ can produce biased similarity scores, which do not reflect the semantic meaning in t, due to focusing only on a subset of the semantically important words in t. Our method remedies this bias by adding a loss term that encourages CLIP to attend to all semantic words in t.

Let S be the set of semantic words in t. Since the textual prompts for image editing are of the format: "a person/ man/ woman with $\{description\}$" or of the format "a $\{description\}$ person/ man/ woman", the set S is considered to

be the words that comprise the description. Our method adds the following explainability-based loss term to the optimization algorithm A:

$$\mathcal{L}_{expl} = -\lambda_{expl} \frac{1}{|S|} \left(\sum_{s \in S} \mathcal{R}_{expl}(s) \right), \tag{3}$$

where λ_{expl} is a hyperparameter. For example, in Fig. 1, the set of semantic words is defined to be: $S = \{\text{“}purple''\text{, “}hair''\}$. This helps the optimization process to favor results where the similarity score is based on the hair color of the subject of the image. As can be seen in the figure, when our loss is not applied, the optimization results in coloring the shirt and the background.

Choosing λ_{expl} Our modified optimization algorithm has an additional hyperparameter λ_{expl}. Since CLIP-based optimization is sensitive to the choice of hyperparameters, it is better to set them based specifically on the input image i, input text t, and generator G. In order to provide an automatic mechanism for choosing λ_{expl}, we consider a range of possible λ_{expl}, and choose the value of λ_{expl} for which the similarity predicted by CLIP for the generated image and the input text is maximal. Note that after applying our method, the similarity scores become more stable, as they consider all semantic tokens in the input.

3.4 Zero-shot text-to-image with spatial conditioning

While the textual descriptions provided to CLIP can include the spatial positioning of objects, images generated by optimizing CLIP similarity score with such texts tend not to follow these spatial restrictions, as shown in Fig 4. We attribute this to the nature of the task CLIP was trained on, which is predicting how similar a given image is to a given text. The existence of matching entities in both inputs is a stronger indication of their similarity than the positions of the entities. This intuition is reflected in the distribution by speech parts (POS) of the explainability scores calculated for CLIP, as shown in the supplementary.

To alleviate this shortcoming of providing spatial positioning with textual description, we add spatial conditioning as an additional input. As far as we can ascertain, CLIP has not been used before for image generation conditioned on spatial masks. The somewhat related task of CLIP-based zero-shot image inpainting was recently successfully performed by [26, 2], who point out that a simple masking of the input image presented to CLIP, such that the similarity score is only predicted based on a specific region of the image, does not guarantee that the generated object will not deviate from the provided region.

Preventing the generator from producing objects outside the designated spatial locations requires applying additional losses on the background or restricting the optimization process such that only the parts of the latent vector that affect the desired region of the image are optimized. These methods limit the spatial conditioning to applications that receive an input image to be used as unaltered background. However, since the explainability maps produced for CLIP indicate the location of the objects, we can effectively limit the location of generated objects using explainability.

Algorithm 1 Obtain IoU loss from masks and image.

Input : (i) $m_1, ..., m_k$- bounding boxes of the objects to be generated, (ii) $t_1, ..., t_k$- textual descriptions of the objects we wish to generate, (iii) C- a pre-trained CLIP model. (iv) the input image i, (v) a threshold T over relevancy scores (vi) *temp* - a temperature for the sigmoid operation (vii) *expl* - the image explainability algorithm for CLIP, which outputs relevance scores for the image patches for each pair of image and text.

Output : $\mathcal{L}_{IoU}$- an explainability-based IoU loss for the input masks $m_1, ..., m_k$, and input image i.

1: $\mathcal{L}_{IoU} \leftarrow 0$
2: *for* $j \in \{1, \ldots, k\}$,:
3: $\quad R_j \leftarrow expl(i, t_j)$
4: $\quad R_j \leftarrow R_j / R_j.max()$
5: $\quad pred\_mask_j \leftarrow sigmoid((R_j - T) * temp)$
6: $\quad intersection \leftarrow \sum_{p \in i}(pred\_mask_j[p] \cdot m_j[p])$
7: $\quad \mathcal{L}_{IoU} \leftarrow \mathcal{L}_{IoU} + \frac{2 * intersection}{\sum_{p \in i} pred\_mask_j[p] + \sum_{p \in i} m_j[p]}$

Our method employs an IoU-inspired loss based on the explainability of the image modality. Alg. 1 describes how we produce the loss $\mathcal{L}_{IoU}$ given the input spatial conditioning masks $m_1, ..., m_k$ and the input image i. For each bounding box m_j and the text t_j describing the object we wish to generate in that location, we generate the explainability for CLIP with the entire image i and text t_j (L.3). This explainability map represents the location in which the object is currently found in the image by CLIP. We then transform the explainability map into a semi-binary mask (L.5) by substracting a threshold value T and passing the output through a sigmoid function with high temperature *temp*. This predicted mask is then used to calculate a Dice Loss with respect to the ground truth object mask (L.7). After calculating the IoU-based loss, we incorporate the similarity-based loss $\mathcal{L} = -\lambda_{expl} \cdot \mathcal{L}_{IoU} - \sum_{j=1}^{k} \text{CLIP}(i, t_j)$, where λ_{expl} is a hyperparameter, and the sum calculates the CLIP similarity between the image and each object we wish to generate, in order to ensure that all objects in $\{t_1, ..., t_k\}$ appear in i. λ_{expl}, T and *temp* are chosen empirically, using examples from the MSCOCO [22] validation set.

4 Experiments

We evaluate our method in various contexts, including one-shot prompt engineering for image classification based on [42], zero-shot text-guided image manipulation based on [28], and zero-shot text-guided image generation with spatial conditioning based on CLIP-guidance for VQGAN [13, 10].

4.1 One-shot prompt engineering

We compare the classification accuracy of CLIP using the prompts optimized with CoOp [42] and with our method, as described in Sec. 3.2. Following [42],

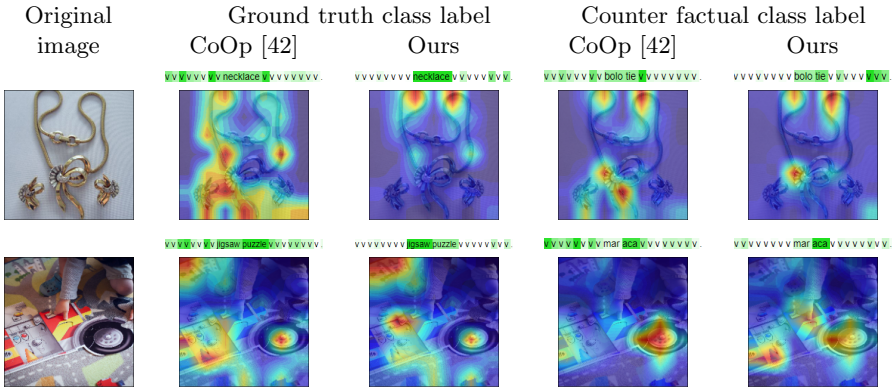

Fig. 2. A qualitative comparison of prompt engineering using CoOp [42] with and without our method on 2 exemplary samples from ImageNetV2 [31]. We present the relevance maps for the ground truth class chosen by our method ("necklace", "jigsaw"), and the counterfactual class chosen by CoOp ("bolo tie", "maraca"). The learned vectors for the prompt are annotated by the letter "v" in the textual explainability maps, since the vectors do not represent actual tokens. As can be seen, for the ground truth classes "necklace" and "jigsaw", our prompts encourage CLIP to focus on the class name in the input text, while CoOp leads CLIP to consider unrelated tokens. This can cause CLIP to produce biased similarity scores based on the engineered prompts.

we evaluate the methods on ImageNet [11] test set, ImageNetV2 [31], ImageNet-Sketch [39], ImageNet-A [18], and Imagenet-R [17].

Following [42], two scenarios are tested: unified prompt engineering and class-specific prompt engineering. In the unified scenario, a single prompt is optimized for all class names. In the class-specific (CSC) case, a different prompt is optimized per class. Note that for all datasets, the prompts are optimized using labeled examples only from the ImageNet training set, in order to test the robustness of the optimized prompts on different ImageNet variations.

For both methods we test different backbones for the visual encoder of CLIP (see Tab. 1), including variations of ViT [12] and of ResNet [16]. Following [42], we optimize a template with $M = 16$ tokens. We also include results for $M = 4$, as it was noted to sometimes achieve superior results on ImageNet.

Two options for positioning the class name tokens in the prompt were reported in [42], with similar outcomes. The first has the class name located in the middle of the prompt, i.e.: $t = v_1, ..., v_8, \{\text{label}\}, v_9, ..., v_{16}$, where $v_1, ..., v_{16}$ are the prompt tokens, and the second has the class name located at the end, i.e.: $t = v_1, ..., v_{16}, \{\text{label}\}$. In the main text we report the results of the former; for the latter, see the supplementary. We use $\lambda_{expl} = 1$ for experiments that use ViT-B/16 as backbone and $\lambda_{expl} = 3$ for all other backbones.

Tab. 1 shows the 1-shot accuracy of CoOp and our method, in addition to 0-shot manual prompt selection and linear probing of the image embedding produced by CLIP, which are the baselines used by CoOp [42]. 2-shot and 4-shot results are

Image backbone		ImageNet		ImageNetV2		INet-Sketch		ImageNet-A		ImageNet-R	
		Unified	CSC	Unified	CSC	Unified	CSC	Unified	CSC	Unified	CSC
ResNet-50	0-shot	58.18	-	51.34	-	**33.32**	-	21.65	-	56.00	-
	LP	21.70	-	17.78	-	5.57	-	0.11	-	0.07	-
	CoOp M=16	54.45	28.40	47.11	23.92	28.12	11.80	19.97	10.39	50.38	26.83
	CoOp M=4	57.63	35.88	50.34	30.49	30.18	16.28	21.43	13.45	53.53	32.06
	Ours M=16	58.13	31.90	51.30	26.82	32.49	13.52	22.12	11.77	**57.73**	29.26
	Ours M=4	**59.05**	**38.79**	**52.33**	**33.58**	32.59	**18.25**	**22.74**	**14.33**	57.15	**34.63**
ResNet-101	0-shot	61.62	-	54.81	-	**38.71**	-	28.05	-	64.38	-
	LP	26.41	-	21.75	-	9.61	-	0.08	-	0.07	-
	CoOp M=16	57.84	33.51	51.25	26.98	33.80	15.78	26.82	14.28	59.02	32.40
	CoOp M=4	60.41	38.96	53.68	33.43	36.71	21.19	27.94	16.91	61.08	40.27
	Ours M=16	61.76	36.01	55.02	31.07	37.96	18.70	29.56	15.97	63.92	36.02
	Ours M=4	**62.31**	**40.77**	**55.65**	**35.18**	38.51	**21.88**	**30.07**	**17.80**	**65.33**	**40.44**
ViT-B/16	0-shot	66.73	-	60.83	-	46.15	-	47.77	-	73.76	-
	LP	32.26	-	27.33	-	16.48	-	0.10	-	0.08	-
	CoOp M=16	63.66	38.86	56.53	33.55	40.96	22.59	43.93	23.30	69.33	42.76
	CoOp M=4	66.93	46.20	60.14	40.28	44.97	28.26	47.44	31.87	72.12	51.16
	Ours M=16	67.09	40.78	60.28	35.25	45.71	23.77	48.29	25.03	74.9	44.43
	Ours M=4	**67.62**	**48.74**	**61.07**	**42.58**	**46.33**	**30.34**	**49.46**	**34.08**	**75.66**	**53.75**
ViT-B/32	0-shot	62.05	-	54.79	-	**40.82**	-	29.57	-	65.99	-
	LP	27.03	-	22.38	-	11.32	-	0.12	-	0.08	-
	CoOp M=16	57.64	33.42	50.24	28.39	35.12	17.63	27.53	13.84	59.46	34.30
	CoOp M=4	61.48	40.66	54.01	34.52	38.26	22.76	29.56	18.58	63.11	41.12
	Ours M=16	62.55	38.63	55.14	33.23	40.40	21.08	31.22	16.8	**67.22**	39.64
	Ours M=4	**63.69**	**42.98**	**55.84**	**37.21**	40.23	**24.26**	**30.78**	**20.48**	66.49	**44.22**

Table 1. 1-shot accuracy (in percentage) of linear probing (LP) and CLIP [29] with prompts produced by the method of [42] (CoOp) or with our explainability-guided variant, with various image backbones. All methods are trained on ImageNet and evaluated on several variants. Unified stands for training a single prompt for all classes, and CSC (class-specific) stands for optimizing a prompt for each class name. Results are averaged over 3 random seeds.

available in the supplementary. As can be seen, both linear probing and CoOp are heavily overfitting and actually achieve significantly lower accuracy than 0-shot results. Using the explainability-based loss, our method is consistently able to improve upon CoOp, leading to higher accuracy across all backbones, all datasets, and both scenarios (unified and CSC).

A Sensitivity analysis for λ_{expl} is presented in Fig. 5, showing that the improvement in accuracy is consistent across a large range of λ_{expl} values. Fig. 2 presents a qualitative comparison of using CoOp with and without our method, see caption for a detailed description.

4.2 Zero-shot text-guided image manipulation

Next, we compare our explainability-based optimization (Sec. 3.3) with the optimization presented in [28]. There are three methods for text-based image editing

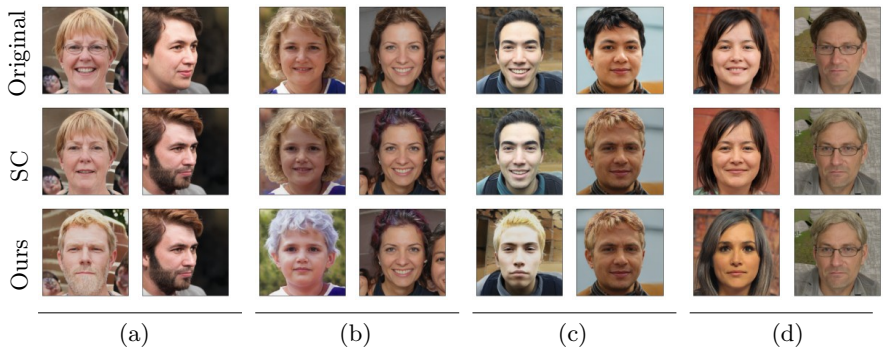

Fig. 3. A qualitative comparison between StyleCLIP (SC) and our method on 4 different textual prompts. (a) "A man with a beard", (b) "A person with purple hair", (c) "A blond man", (d) "A person with grey hair". For each prompt we present examples where StyleCLIP is successful (right column), and unsuccessful (left column). For the failure cases, the optimization in StyleCLIP hardly modifies the original image, leading to a high identity preservation score when no semantic change was applied. When StyleCLIP is successful, our method produces similar or identical results.

using StyleGAN [19] presented by [28] - latent optimization, mapper training, and global directions extraction. We focus on latent optimization, since our focus is on zero-shot methods and the other two methods employ additional training. As described in Sec. 3.3, we add the explainability-based loss from Eq. 3 to the optimization algorithm of [28]. We choose the set of hyperparameters for our explainability-based loss from the set: $\lambda_{expl} = \{0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5\}$, and use the best value for λ_{expl} according to the highest CLIP similarity score.

Since the optimization in [28] requires a different hyperparameter setting for each prompt, we select prompts that appear in the paper or official code, and use the same hyperparameters (in other words, we do not manually select the hyperparameters for our method). Next, we choose 20 random seeds to be used across all our experiments to generate the images to be edited, i. For each image i, and text prompt t we produce the edited image with StyleCLIP's optimization, and with our modified optimization.

For evaluation, we extract all examples where our method produces a different output than StyleCLIP, i.e., all cases where the automatic procedure selected $\lambda_{expl} \neq 0$, and conduct a user study among 46 users. Users were asked to evaluate each manipulation by the similarity between the manipulated image and the input prompt t and by the loss of identity between the manipulated image and the original image i, both on a scale of $1 - 5$ (higher is better).

Fig. 3 presents sample results from our user study (See the supplementary for full results). Notice that for challenging manipulations, such as using the prompt "a man with a beard" on a woman, StyleCLIP tends to leave the input image i almost unchanged. In contrast, in many of these cases, our method compels the optimization to reach a solution that fits the required semantic edit. We present the results of the user study for each prompt in Tab. 2 (see results with

	A man with a beard		A person with purple hair		A blond man		A person with grey hair	
Method	Quality	Identity	Quality	Identity	Quality	Identity	Quality	Identity
SC	2.92	**3.61**	1.17	**4.13**	3.93	**3.67**	2.59	**3.82**
Ours	**4.28**	2.23	**2.29**	3.51	**4.28**	2.63	**3.27**	3.10

Table 2. A user study comparing text-based image editing with StyleCLIP (SC) and our method on 4 different textual prompts: "A man with a beard", "A person with purple hair", "A blond man", "A person with grey hair". Quality refers to the similarity between the prompt and the manipulation; Identity refers to the identity preservation of the manipulation. Scores are averaged across 20 random seeds, on a scale of 1-5 (higher is better).

standard deviation in the supplementary). As can be seen, our method produces results that are consistently rated by users as more similar to the target text. However, StyleCLIP, which, as can be seen in Fig. 3, often leaves the images unchanged, obtains a higher identity preservation score. Evidently, the gap in the identity score is much bigger for the prompts "A **man** with a beard" and "A blond **man**". These prompts modify the gender of the subject of the image i, thereby requiring a more substantial identity change.

4.3 Zero-shot text-to-image with spatial conditioning

We use CLIP-guided VQGAN as implemented by [10]. Since, as far as we can ascertain, there is no previous literature on zero-shot CLIP-guided text-to-image generation with spatial conditioning on the location of the generated objects, we use two variations of a similarity-based CLIP loss to create baselines without explainability conditioning. The first baseline employs the loss $\mathcal{L}_{masked} = \sum_{t\in\{t_1,\ldots,t_k\}} \sum_{m\in\{m_1,\ldots,m_k\}} -\text{CLIP}(i_m, t)$, where i_m is the image i masked according to bounding box m, i.e. for each mask m, we black out all pixels outside m, in order to ensure that the objects identified by CLIP reside within the bounding boxes, and t is a prompt of the form "a photo of {label}" where "label" is the target class to be generated in bounding box m. This masking technique has also been employed in previous works [26, 2], for CLIP-guided image inpainting. The second similarity-based baseline we consider employs the loss $\mathcal{L}_{masked}$ in addition to the similarity loss in the unmasked image $\mathcal{L}_{similarity} = \sum_{t\in\{t_1,\ldots,t_k\}} -\text{CLIP}(i, t)$. This baseline uses the loss: $\mathcal{L} = \mathcal{L}_{similarity} + \mathcal{L}_{masked}$, which considers both the information inside the bounding boxes and the information in the entire image.

As mentioned in Sec. 3.4, since a simple similarity-based loss has no spatial restrictions, the baselines produce objects outside the input bounding box (see Fig. 4), while our loss produces objects within the bounding box, thanks to spatial conditioning based on explainability. Moreover, the examples in Fig. 4 demonstrate the ability of our method to generate images in a variety of cases, including multiple bounding boxes with varying heights and widths (see the supplementary for additional examples and visualizations of the explainability

Fig. 4. A qualitative comparison between the two similarity-based baselines and our method for CLIP-guided zero-shot text-based image generation with spatial conditioning. Textual conditioning refers to specifying the spatial positioning of objects within the text prompts, for example "a vase on a table". Additional examples are presented in the supplementary.

maps). As smaller bounding boxes require stronger supervision, we set λ_{expl_i} for object i to be $\lambda_{expl_i} = \frac{0.15}{\sqrt{r(m_i)}}$, where m_i is the bounding box assigned to object i and $r(m_i)$ is the ratio between the area of the mask and the area of the entire image. The threshold T is set to 0.1 and $temp$ is set to 20.

In order to provide quantitative metrics for our spatially conditioned text-to-image generation, we use the validation set from MSCOCO [22] which contains bounding boxes for each object in the image. In order to ensure a varying number and size of objects, while maintaining enough background to allow object-free

	Similarity-based	Similarity-based 2	Ours
Precision	46.4	26.9	**71.7**
Recall	48.3	30.5	**63.4**
F1	40.5	24.28	**62.6**
AP	8	5.4	**26.2**
AR	21.6	19	**40**
$AP_{0.5}$	18	15.4	**56.5**

Table 3. Precision, recall, F1, average precision, and average recall for spatially conditioned image generation with our method, and two similarity-based baselines (results in percentage). Metrics were averaged across 100 random samples from the MSCOCO [22] validation set and four random seeds. Average precision and average recall are calculated using DETR [5].

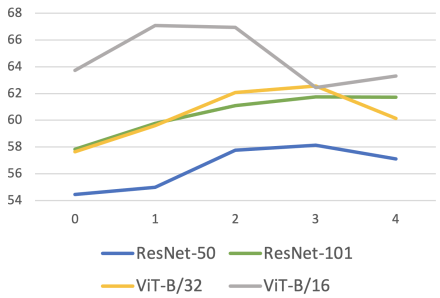

Fig. 5. 1-shot accuracy (in percentage) on the ImageNet test set for different choices of λ_{expl} for all visual backbones of CLIP. The accuracy achieved by the baselines is denoted as $\lambda_{expl} = 0$.

generation, which is challenging for CLIP-guided VQGAN, we filter the layout as follows: we keep the k largest bounding boxes whose commutative area is less than 50% of the image, where adding the next largest bounding box would result in occupying more than 50% of the image. By focusing on the largest objects we also help ensure that the size of each bounding box suffices for the CLIP encoder. For our first experiment, we sample 100 MSCOCO images at random, and use our method and the similarity-based baselines to generate images corresponding to the annotated spatial layout. We then produce an explainability map for each text description t, as described in Alg. 1 (L.2-3). We use these maps as soft semantic segmentation, binarize them using thresholds produced with Otsu's method [27], and calculate the precision, recall, and F1 scores of the binarized maps with the ground truth bounding boxes $m_1, .., m_k$. Note that both precision and recall are limited and cannot reach 100% due to the square shape of the bounding boxes, which is not suited to non-square objects. Precision is also limited because images often contain more than one instance of a specific class, leading to a high explainability score for the other occurrences as well. As can be seen in Tab. 3, our method significantly outperforms the baselines.

Next, we use object detection to evaluate the quality of the generated objects, as well as the overlap between their location and the target spatial condition. DETR [5] is used to produce bounding boxes for each object. These bounding boxes are evaluated against the input spatial conditioning masks using the average precision and average recall scores. As can be seen in Tab. 3, our method greatly outperforms the baselines in this evaluation as well, implying that the explainability signal is indeed indicative enough to enforce spatial restrictions over an image.

5 Discussion

In our experiments, we presented a generic application of explainability to improve classification, image editing, and image synthesis. There are also specific situations in which a limited view of the input is detrimental and where explainability can help ensure a more even distribution of information pooling. One such case, studied in the supplementary, is that of compound nouns, e.g. "apple juice" or "blueberry muffin". As we show, state-of-the-art zero-shot text-to-image generation engines might overly emphasize or ignore some of the textual input, leading to misinterpretation of the text. The method we present for equalizing the contributions to avoid such neglect not only leads to considerably better image outputs for such cases, but also slightly improves the FID score for other sentences. See the supplementary for full details of the method implementation, visual examples, and the results of a user study conducted against results obtained with a state-of-the-art method. In order to demonstrate the wide applicability of our approach, we have modified multiple zero-shot and one-shot approaches. While the baseline approaches are impressive, we do note that they are not yet ready to replace supervised methods. Prompt engineering is not yet competitive with supervised methods, CLIP-guided VQGAN often generates substandard images, and StyleCLIP optimization method often fails and requires different parameters for each prompt. Therefore, other signals need to be considered to allow zero-shot applications to compete against fully supervised ones. Explainability, as we show, is an example of such beneficial signal.

6 Conclusions

While explainability methods are constantly improving, their use as a feedback mechanism to improve classification or generation methods is still relatively unexplored. As far as we can ascertain, their utilization as such a building block is currently limited to weakly supervised segmentation [41, 7]. In this work, we show how explainability can help overcome the neglect problem of bi-modal transformers, which have become a cornerstone in the current rapid evolution of zero-shot methods. We demonstrate how preventing neglect, as reflected through the lens of the explainability score, helps improve one-shot classification, zero-shot image editing, and zero-shot layout-conditioned image generation. In the first two domains, neglect is prevented in the text domain, while in the latter, the constraint on the heatmap is placed in the image domain.

Acknowledgments This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant ERC CoG 725974). We thank Ariel Landau for his assistance.

References

1. Abnar, S., Zuidema, W.: Quantifying attention flow in transformers. arXiv preprint arXiv:2005.00928 (2020)
2. Bau, D., Andonian, A., Cui, A., Park, Y., Jahanian, A., Oliva, A., Torralba, A.: Paint by word. arXiv preprint arXiv:2103.10951 (2021)
3. Berg, T., Forsyth, D.: Animals on the web. In: CVPR (2006)
4. Binder, A., Montavon, G., Lapuschkin, S., Müller, K.R., Samek, W.: Layer-wise relevance propagation for neural networks with local renormalization layers. In: International Conference on Artificial Neural Networks. pp. 63–71. Springer (2016)
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. arXiv preprint arXiv:2005.12872 (2020)
6. Chefer, H., Benaim, S., Paiss, R., Wolf, L.: Image-based clip-guided essence transfer (2021)
7. Chefer, H., Gur, S., Wolf, L.: Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 397–406 (October 2021)
8. Chefer, H., Gur, S., Wolf, L.: Transformer interpretability beyond attention visualization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 782–791 (June 2021)
9. Chen, X., Gupta, A.K.: Webly supervised learning of convolutional networks. 2015 IEEE International Conference on Computer Vision (ICCV) pp. 1431–1439 (2015)
10. Crowson, K.: Vqgan+clip. https://colab.research.google.com/drive/1L8oL-vLJXVcRzCFbPwOoMkPKJ8-aYdPN (2021)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition pp. 248–255 (2009)
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
13. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12873–12883 (June 2021)
14. Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: Learning object categories from internet image searches. Proceedings of the IEEE **98**(8), 1453–1466 (2010). https://doi.org/10.1109/JPROC.2010.2048990
15. Gal, R., Patashnik, O., Maron, H., Chechik, G., Cohen-Or, D.: Stylegan-nada: Clip-guided domain adaptation of image generators (2021)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
17. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., Gilmer, J.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8340–8349 (October 2021)
18. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15262–15271 (June 2021)

19. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: CVPR. pp. 8110–8119 (2020)
20. Kim, G., Ye, J.C.: Diffusionclip: Text-guided image manipulation using diffusion models (2021)
21. Li, A., Jabri, A., Joulin, A., van der Maaten, L.: Learning visual n-grams from web data. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
22. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
23. Liu, X., Gong, C., Wu, L., Zhang, S., Su, H., Liu, Q.: Fusedream: Training-free text-to-image generation with improved clip+gan space optimization. ArXiv **abs/2112.01573** (2021)
24. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems. pp. 4765–4774 (2017)
25. Michel, O.J., Bar-On, R., Liu, R., Benaim, S., Hanocka, R.: Text2mesh: Text-driven neural stylization for meshes. ArXiv **abs/2112.03221** (2021)
26. Omri Avrahami, D.L., Friedn, O.: Blended diffusion for text-driven editing of natural images. arXiv preprint arxiv:2111.14818 (2021)
27. Otsu, N.: A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics **9**(1), 62–66 (1979)
28. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: Styleclip: Text-driven manipulation of stylegan imagery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2085–2094 (2021)
29. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. arXiv preprint arXiv:2103.00020 (2021)
30. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
31. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do ImageNet classifiers generalize to ImageNet? In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 5389–5400. PMLR (09–15 Jun 2019), https://proceedings.mlr.press/v97/recht19a.html
32. Rubinstein, M., Joulin, A., Kopf, J., Liu, C.: Unsupervised joint object discovery and segmentation in internet images. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1939–1946 (2013). https://doi.org/10.1109/CVPR.2013.253
33. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
34. Shrikumar, A., Greenside, P., Shcherbina, A., Kundaje, A.: Not just a black box: Learning important features through propagating activation differences. arXiv preprint arXiv:1605.01713 (2016)
35. Tang, K., Joulin, A., Li, L.J., Fei-Fei, L.: Co-localization in real-world images. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1464–1471 (2014). https://doi.org/10.1109/CVPR.2014.190
36. Tewel, Y., Shalev, Y., Schwartz, I., Wolf, L.: Zero-shot image-to-text generation for visual-semantic arithmetic. CoRR **abs/2111.14447** (2021), https://arxiv.org/abs/2111.14447

37. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
38. Vijayanarasimhan, S., Grauman, K.: Keywords to visual categories: Multiple-instance learning forweakly supervised object categorization. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (2008). https://doi.org/10.1109/CVPR.2008.4587632
39. Wang, H., Ge, S., Lipton, Z., Xing, E.P.: Learning robust global representations by penalizing local predictive power. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019), https://proceedings.neurips.cc/paper/2019/file/3eefceb8087e964f89c2d59e8a249915-Paper.pdf
40. Wang, X.J., Zhang, L., Li, X., Ma, W.Y.: Annotating images by mining image search results. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(11), 1919–1932 (2008). https://doi.org/10.1109/TPAMI.2008.127
41. Zabari, N., Hoshen, Y.: Semantic segmentation in-the-wild without seeing any segmentation examples. ArXiv **abs/2112.03185** (2021)
42. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. arXiv preprint arXiv:2109.01134 (2021)
43. Zhu, P., Abdal, R., Femiani, J.C., Wonka, P.: Mind the gap: Domain gap control for single shot domain adaptation for generative adversarial networks. ArXiv **abs/2110.08398** (2021)