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

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

The Blavatnik School of Computer Science, Tel Aviv University

1 Explainability Method

We create relevance maps for the text tokens (denoted by $\mathbf{R}^{tt}$), and for the image tokens (denoted by $\mathbf{R}^{ii}$) following the method presented in [5]. We initialize the maps as follows:

$$\mathbf{R}^{ii} = \mathbb{I}^{i \times i}, \quad \mathbf{R}^{tt} = \mathbb{I}^{t \times t} \tag{1}$$

Next, we update the relevance maps by a forward pass on the attention layers. We use gradients in order to average across the attention heads, as done in [5]:

$$\bar{\mathbf{A}} = \mathbb{E}_h((\nabla \mathbf{A} \odot \mathbf{A})^+) \tag{2}$$

where $\odot$ is the Hadamard product, $\nabla \mathbf{A} := \frac{\partial s_{t,i}}{\partial \mathbf{A}}$ for $s_{t,i}$ which is the the similarity score computed by CLIP for the text prompt t with the image i, and $\mathbb{E}_h$ is the mean across the heads dimension. Note that the propagation of gradients by the similarity score allows us to obtain explainability scores for the text that are specific to the input image, i.e. different images induce different explainability scores for each textual token and vice versa.

Finally, to incorporate each layer's explainability map to the accumulated relevancy maps, we use the propagation rule presented in [5] for self-attention layers:

$$\mathbf{R}^{ii} \leftarrow \mathbf{R}^{ii} + \bar{\mathbf{A}}_{\mathbf{i}} \cdot \mathbf{R}^{ii} \tag{3}$$

$$\mathbf{R}^{tt} \leftarrow \mathbf{R}^{tt} + \bar{\mathbf{A}}_{\mathbf{t}} \cdot \mathbf{R}^{tt} \tag{4}$$

where $\bar{\mathbf{A}}_{\mathbf{i}}$, $\bar{\mathbf{A}}_{\mathbf{t}}$ are attention relevance maps for image self-attention layers and text attention layers, respectively, which were calculated using Eq. 2.

To obtain relevance per each text token, we observe that CLIP uses the *eot* token as a classification token, thus we simply use the row of $\mathbf{R}^{tt}$ that corresponds to the *eot* token.

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

Dataset	ResNet-50		ResNet-101		ViT-B/16		ViT-B/32	
	CoOp	Ours	CoOp	Ours	CoOp	Ours	CoOp	Ours
ImageNet	53.41	**57.78**	56.87	**61.12**	63.63	**66.19**	58.23	**61.38**
ImageNetV2	46.65	**51.20**	50.17	**54.54**	56.69	**58.40**	50.93	**53.82**
ImageNet-Sketch	27.90	**32.41**	34.43	**37.58**	41.70	**44.66**	35.33	**38.75**
ImageNet-A	20.45	**21.89**	26.63	**28.74**	45.58	**46.03**	27.92	**30.6**
ImageNet-R	50.31	**57.16**	58.93	**63.95**	71.24	**73.77**	61.18	**65.49**

Table 1. 1-shot accuracy (in percentage) of CLIP [10] with prompts produced by the method of [12] (CoOp) and by our explainability-guided variant, with class name tokens positioned at the end of the prompts. Results are averaged over 3 random seeds.

2 Additional prompt engineering results

We present additional results for few-shot classification, which were left out of the main text for brevity. We do not present results for the class-specific configuration due to the required computational resources.

2.1 Prompt engineering with class name at the end of prompt

As mentioned in Sec. 4.1 in the main text, Zhou et al. [12] proposed two options for positioning the class name tokens in the optimized prompts, which achieved similar results. The first has the class name positioned in the middle of the prompt, i.e.: t = v1,...,v8,label,v9,...,v16, where v1,...,v16 are the prompt tokens, and the second has the class name located at the end, i.e.: t = v1,...,v16,label. The results of our explainability-based method for the middle positioning variant are reported in the main text. The results for the end positioning variant are presented in Tab. 1. As can be seen, our method consistently outperforms the original method of [12] with the end positioning configuration, similar to the middle positioning one.

2.2 Few-shot prompt engineering

Although this work focuses on the 1-shot scenario for prompt engineering, we report the results for 2-shot and 4-shot optimization as well. Tab. 2 and 3 present the accuracy achieved with the method proposed in [12] and with our method for the unified prompt configuration, where a single prompt is optimized for all class names. As can be seen in Tab. 2, our method consistently outperforms the original method of [12] with a significant margin across the different datasets and backbones in the 2-shot scenario. A similar situation occurs in the 4-shot scenario, as can be seen in Tab. 3. Note that for some of the visual backbones, the accuracy our method achieves for 1-shot optimization surpasses the accuracy achieved by the original method in the 2-shot scenario, and the results for our

Dataset	ResNet-50		ResNet-101		ViT-B/16		ViT-B/32	
	CoOp	Ours	CoOp	Ours	CoOp	Ours	CoOp	Ours
ImageNet	56.70	**58.42**	60.59	**62.71**	65.94	**67.72**	60.59	**62.71**
ImageNetV2	49.99	**51.99**	52.85	**54.95**	58.48	**60.87**	52.66	**55.10**
ImageNet-Sketch	29.31	**31.46**	36.03	**37.95**	43.06	**45.79**	37.18	**39.70**
ImageNet-A	21.85	**22.19**	26.98	**28.83**	45.69	**46.96**	28.68	**30.87**
ImageNet-R	52.03	**56.34**	60.73	**63.54**	71.15	**74.22**	61.84	**64.92**

Table 2. 2-shot accuracy (in percentage) of CLIP [10] with prompts produced by the method [12] (CoOp) and by our explainability-guided variant. Results are averaged over 3 random seeds.

Dataset	ResNet-50		ResNet-101		ViT-B/16		ViT-B/32	
	CoOp	Ours	CoOp	Ours	CoOp	Ours	CoOp	Ours
ImageNet	59.50	**59.87**	62.84	**63.59**	68.20	**68.90**	63.12	**64.14**
ImageNetV2	52.04	**52.22**	55.31	**56.25**	61.39	**61.85**	54.72	**56.57**
ImageNet-Sketch	30.89	**32.45**	36.97	**38.84**	44.40	**45.92**	38.31	**40.18**
ImageNet-A	21.83	**22.48**	28.37	**29.93**	46.93	**48.23**	29.50	**31.14**
ImageNet-R	54.03	**55.82**	60.61	**63.78**	71.36	**74.13**	62.47	**65.68**

Table 3. 4-shot accuracy (in percentage) of CLIP [10] with prompts produced by the method of [12] (CoOp) and by our explainability-guided variant. Results are averaged over 3 random seeds.

method in the 2-shot scenario are very close to the results of the original method in the 4-shot scenario. Since each i-shot, for $i \in \{2, 4\}$ duplicates the size of the training set compared to $(i-1)$-shot, this is a strong indication for the effectiveness of our method in improving generalization.

3 Relevance scores distribution over different POS

In order to understand the limitations of CLIP-guided optimization in general, we study the importance of different speech parts (POS) to the prediction delivered by CLIP. We calculate the textual explainability scores for all matching pairs of image and caption in MSCOCO [7] validation set. The explainability scores for each caption are divided by the maximal explainability score in it, to allow comparison of the relevancy of words between different sentences. We extract the POS of each caption using the part-of-speech tagging architecture of [2] with the Flair framework [1], and average the relevance scores of each POS over the entire MSCOCO [7] validation set. Fig. 1 shows the average relevance

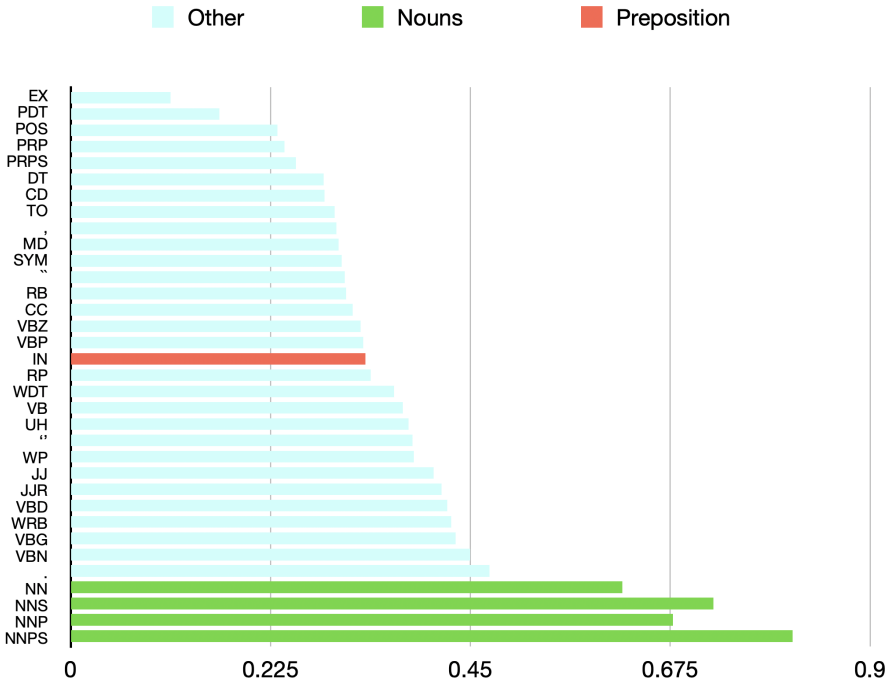

Fig. 1. Average textual explainability score for different speech parts (POS), calculated over the matching pairs of text and image in MSCOCO [7] validation set. The POS that describe entities are colored in green, and preposition is colored in red. As can be seen, when predicting that truly matching image and caption are similar, the nouns (objects) are most relevant to the prediction, significantly more that the spatial positioning.

score of different speech parts, for all speech parts that appear at least 20 times in the data.

As can be seen in Fig. 1 CLIP bases its similarity scores on the nouns significantly more than any other speech part in the text, including prepositions (IN) and adjectives (JJ), meaning that the existence of a given object in both the caption and the image is more important for the similarity score prediction than its detailed attributes or spatial position. This observation explains why CLIP-guided optimizations often fail to follow spatial positions described in the input text prompt.

4 Text-guided image generation

We describe a method for text-guided image generation that is similar in spirit to the methods presented in the main text. For brevity, and since the benefit of the proposed improvement is most apparent in the specific case of compound

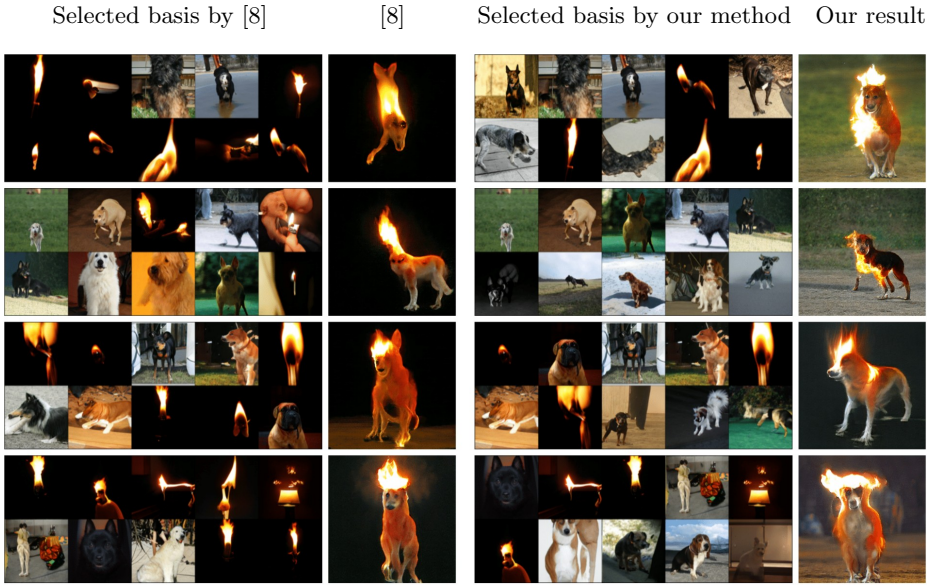

Fig. 2. Generated images conditioned on the prompt "a photo of a flaming dog" using FuseDream [8] with and without our method. The word "flaming" is very dominant and the original FuseDream basis selection results with mostly images of flames. This unbalanced basis limits the ability of the optimization process to generate the dog and leads to a small variation both in the background and the dog itself.

nouns, we exclude the results from the main text. Moreover, unlike the loss-based approach used in the paper, the intervention here is only in the retrieval mechanism used for the basis selection stage of the generative scheme.

The FuseDream method [8] improves upon a direct application of BigGAN [3] for CLIP-guided text-based image generation. FuseDream employs a modified CLIP score based on image augmentations, as follows:

$$\text{AUGCLIP}(t, i) = \mathbb{E}_{i' \sim \pi(\cdot|i)}[\text{CLIP}(t, i')], \tag{5}$$

where i' is a random augmentation of i drawn from the distribution $\pi(\cdot|i)$. The data augmentations are adopted from [11].

Since the optimization process is essentially a traversal in the latent space of a pretrained BigGAN, aimed to locate the latent vector from which BigGAN generates an image that matches the text prompt, a key contribution of [8] regards the initialization of this traversal. Instead of initializing the optimization process with a randomly sampled latent vector, the FuseDream method randomly generates M images using BigGAN, and selects the vectors in the BigGAN latent space that generate the k images with the highest AUGCLIP score as a basis B. This basis selection phase allows for shorter and easier navigation of the latent space, as it is initialized closer to the target vector to be reached. Let $B = \{v_1, ..., v_k\}$ be the selected basis (where $v_1, ..., v_k$ are latent vectors). The

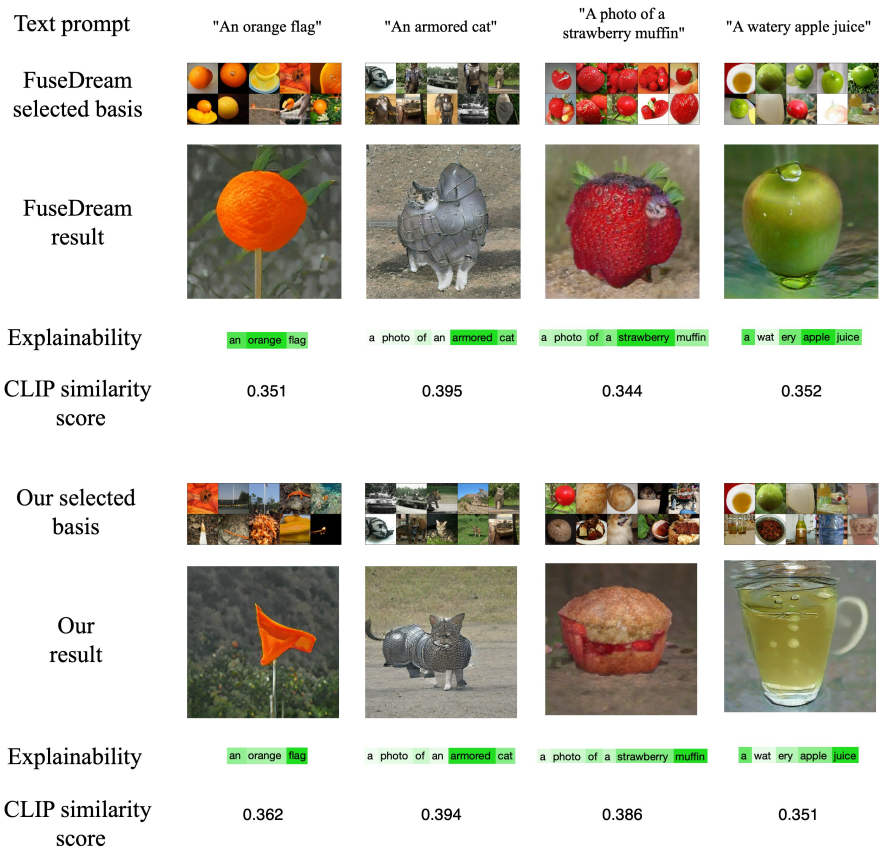

Fig. 3. Examples of selected bases and resulting generated images with FuseDream [8] with and without our explainability guidance, on prompts with compound nouns presented in the user study. The basis selection of [8] results in basis images that severely neglect relevant nouns from the prompts such as flag, cat, juice, and muffin, as reflected in both the images and their corresponding explainability scores. In contrast, our method selects basis vectors that better correspond to the input prompts, resulting in generated images that follow the semantic meaning of the texts.

FuseDream method optimizes the following:

$$\max_{\{\epsilon_i, w_i\}_{i=1}^{k}} \text{AUGCLIP}\left(t, g\left(\sum_{j=1}^{k} w_i \epsilon_i\right)\right), \tag{6}$$

where g is the BigGAN generator, the vectors are initialized to be the basis vectors: $\epsilon_j = v_j$, and the coefficients are initialized as: $w_j = \frac{1}{k}$. The final output image after the optimization is: $i_{out} = g\left(\sum_{j=1}^{k} w_i \epsilon_i\right)$. Since the basis selection relies solely on the similarity score predicted by CLIP, it can exhibit a neglecting

behavior, resulting in a retrieved basis that is unbalanced and fails to represent the semantic meaning of the input text. An example can be seen in Fig. 3, for the phrase "a photo of a strawberry muffin", all images chosen as the basis of FuseDream feature strawberries, which leads to a sub-optimal initialization of the optimization process, resulting in output images that are shaped like a strawberry, rather than a muffin. This can be attributed to the fact that for the images corresponding to strawberries, CLIP predicts a similarity score that is based mostly on the word "strawberry", disregarding the word "muffin". Fig. 2 presents another example, for the basis produced for the prompt "a photo of a flaming dog", in which the word "flaming" is emphasized over "dog". In order to overcome this sensitivity, our method accounts for the explainability scores of the produced similarity, instead of simply considering the pure similarity scores, by adding $\mathcal{L}_{expl}$ from Eq. 4 in the main text to the AUGCLIP score to select the basis vectors. This ensures that the basis vectors indeed reflect the entire semantic content of the textual prompt.

Choosing the set of semantic words S: Eq. 4 in the main text uses a set of semantic words S to ensure that the similarity score predicted by CLIP is based on the true semantic meaning of the input text. In order to produce the set S automatically, our method uses a part-of-speech tagging architecture [2] with the Flair framework [1] to automatically extract all words that correspond to nouns in the input text prompt t. Next, the FuseDream [8] method is used to generate an image i by the description t, and a relevance score $\mathcal{R}_{expl}$ is computed for each word in t w.r.t. the generated image i, as described in Sec. 3.1 in the main text. The set S is defined as follows:

$$S = \{t_i \in t_{noun} \text{ s.t. } \mathcal{R}_{expl}(t_i) < 0.7\}, \tag{7}$$

where t_{noun} is the set of words in t that are classified by the part-of-speech tagger as nouns. Therefore, the loss in Eq. 4 in the main text emphasizes all nouns in the input prompt with a relevance score lower than 0.7 for the produced image i. Intuitively, each noun represents an object that should appear in the image i, therefore nouns that have a low relevance score either do not appear or appear only partially in the image i. The loss in Eq. 4 in the main text ensures that the neglected objects of i will appear once it is applied to the modified basis selection. In our experiments, we set: λ_{expl}=0.1.

4.1 FuseDream experiments

We conduct all our experiments with the default setting from the FuseDream [8] code base, using a BigGAN [3] generator for 512 resolution images, with 10 basis vectors for each image generation. To evaluate the visual quality of both methods, we conduct a user study, with 46 participants. The study presents the users with 53 textual prompts, for which we generate corresponding images with FuseDream [8], and with our modified method described above. Users are asked to choose the image that corresponds best to the textual description or mark both as equally successful. 46 of the presented prompts are the visual examples presented in [8], so as to compare our method against the most visually

Basis size	FuseDream	Ours
M=5	21.26	**20.66**
M=10	24.67	**23.96**

Table 4. FID [6] (lower is better) calculated on images generated with FuseDream [8] and with our method according to 30,000 randomly sampled prompts from the MSCOCO validation set. Both FuseDream and our method use a pretrained BigGAN [3] generator for 512 resolution images. Our method slightly improves the results of FuseDream.

pleasing results by FuseDream. We focus on the examples where our method produces a different result than FuseDream, meaning the set S contains at least one word. The other 7 prompts are prompts containing compound nouns, such as "strawberry muffin" or "orange kimono", as we found these cases tend to exhibit neglect by CLIP. For each example, we use the majority of the answers to determine which method produced the best image. In 33 of the 53 images (62.26%) the participants ruled that our method produced an image that corresponds better to the input text. Additionally, the average ratings for the 7 prompts that contain compound nouns show that in 86.6% of the cases, users voted for our method as producing the results most compatible with the textual descriptions. See Fig. 3 for a full comparison between the selected basis and the final images produced by our method and FuseDream for 4 of the 7 prompts containing compound nouns. As can be seen, our method selects basis images that correspond better to the target semantic prompt; therefore it generates images that reflect the textual descriptions.

Next, following the metrics presented in [8], we present the Fréchet inception distance (FID) [6] on a subset of 30,000 randomly sampled prompts from the MSCOCO [7] validation set. As can be seen in Tab 4, our method slightly improves the FID score over FuseDream, indicating that, in accordance with the user study, our method either preserves the successful results of FuseDream, or improves the produced images, in cases where the similarity-based optimization fails to capture the entire textual description

5 StyleCLIP user study results

Tab. 5 presents the StyleCLIP user study results with standard deviation per metric. Notice that standard deviations tend to be high since some of the manipulations performed by both methods fail, resulting in low quality scores. As can be seen, our standard deviation for the quality score is consistently lower than StyleCLIP's (with the exception of prompt (c) where both are very similar), indicating that our manipulations are more stable. Fig. 4, 5, 6, 7 present the seeds from the StyleCLIP user study where our method generates a different result than that of StyleCLIP, out of the 20 random seeds used for the study. For the other seeds, our method produces the same result as StyleCLIP ($\lambda_{expl} = 0$). As

Method	Quality	Identity
SC	2.92 (± 1.86)	**3.61** (± 0.68)
Ours	**4.28** (± 0.54)	2.23 (± 0.72)

(a)

Method	Quality	Identity
SC	1.17 (± 0.18)	**4.13** (± 0.27)
Ours	**2.29** (± 1.39)	3.51 (± 0.8)

(b)

Method	Quality	Identity
SC	3.93 (± 1.40)	**3.67** (± 0.80)
Ours	**4.28** (± 0.50)	2.63 (± 0.86)

(c)

Method	Quality	Identity
SC	2.59 (± 1.80)	**3.82** (± 0.78)
Ours	**3.27**(± 1.51)	3.10(± 1.01)

(d)

Table 5. Results of the user study comparing text-based image editing with 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". 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). Notice that standard deviations tend to be high since some of the manipulations preformed by both methods fail, resulting in low quality scores.

mentioned in the main text, for prompts that entail a change of identity such as "a blond *man*" and "a *man* with a beard", our method causes a more significant identity change in accordance with the prompt.

6 Zero-shot text to image generation with spatial conditioning

6.1 Parameter sensitivity for λ_{expl}, T and *temp*

The sensitivity of our method for spatially conditioned image generation to its hyperparameters λ_{expl}, T and *temp* is studied in Tab. 6, 7, and 8 respectively. The temperature *temp* is used along with a *sigmoid* function to transform the continuous and normalized explainabiltiy scores into semi-binarized map. As can be seen in Tab. 6, setting $temp = 1$ reduces all DETR metrics significantly. However, values in a wide range of 10-40 all produce reasonable results.

As can bee seen in Tab. 7 and 8, our method works well with different values for λ_{expl} and T, resulting with significantly better metrics than the baselines.

6.2 Zero-shot text to image generation with spatial conditioning visualizations

Fig. 8, 9 present additional examples of our method for zero-shot text to image generation with spatial conditioning. Fig. 10 presents the generated images along with their explainability maps. Our explainability guidance enforces that

Fig. 4. Results of the StyleCLIP [9] user study on the prompt "A man with a beard." The examples above represent the seeds where our method generates a different result than that of StyleCLIP, out of the 20 random seeds selected for the study. For the other seeds, our method produces the same result as StyleCLIP ($\lambda_{expl} = 0$).

the objects remain within the provided bounding boxes, and the relevancy maps demonstrate the effectiveness of the explainability method in detecting the objects within each box. As can be seen, for most cases, our method successfully generates the images such that each object is contained within its designated bounding box, while the similarity-based baselines tend to deviate from the pro-

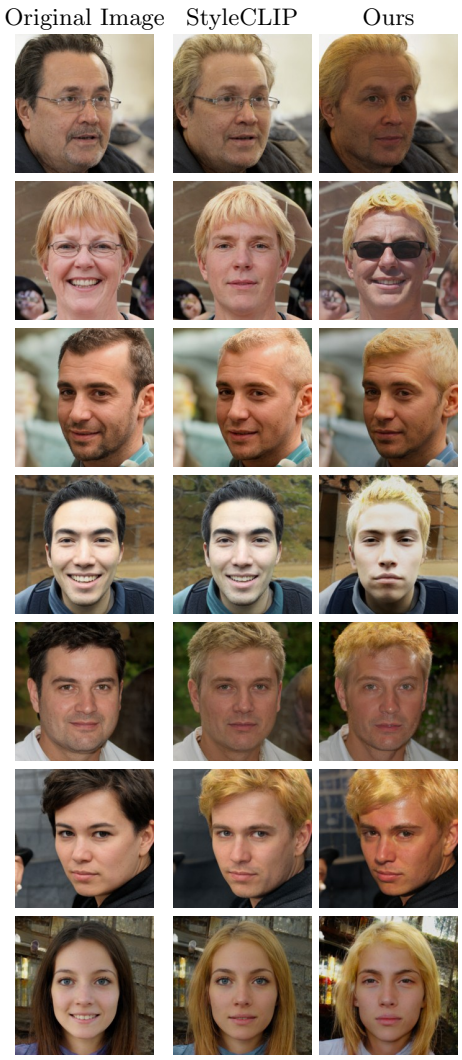

Fig. 5. Results of the StyleCLIP [9] user study on the prompt "A blond man." The examples above represent the seeds where our method generates a different result than that of StyleCLIP, out of the 20 random seeds selected for the study. For the other seeds, our method produces the same result as StyleCLIP ($\lambda_{expl} = 0$).

vided bounding boxes, which is also reflected in artifacts in the explainability maps.

Fig. 6. Results of the StyleCLIP [9] user study on the prompt "A person with purple hair." The examples above represent the seeds where our method generates a different result than that of StyleCLIP, out of the 20 random seeds selected for the study. For the other seeds, our method produces the same result as StyleCLIP ($\lambda_{expl} = 0$).

temp	Precision	Recall	F1	AP	AR	$AP_{0.5}$
1	52.1	72.8	53.6	8.4	21.3	23.4
10	67.5	**76.4**	**66.7**	23.5	38.6	49.8
20*	71.7	63.4	62.6	**26.2**	**40.0**	**56.5**
30	**72.6**	53.0	56.8	23.3	36.0	52.2
40	72.4	46.6	52.1	19.0	34.1	44.8

Table 6. Precision, recall, F1, average precision, and average recall for spatially conditioned image generation with our method, with different values for the hyperparameter *temp*. Metrics are averaged across 100 random samples from the MSCOCO [7] validation set and four random seeds. Average precision and average recall are calculated using DETR [4]. * The value used for our method

Fig. 7. Results of the StyleCLIP [9] user study on the prompt "A person with grey hair." The examples above represent the seeds where our method generates a different result than that of StyleCLIP, out of the 20 random seeds selected for the study. For the other seeds, our method produces the same result as StyleCLIP ($\lambda_{expl} = 0$).

T	Precision	Recall	F1	AP	AR	$AP_{0.5}$
0.05	68.8	**66.7**	62.4	20.8	34.8	44.9
0.1*	71.7	63.4	**62.6**	**26.2**	**40.0**	**56.5**
0.2	73.4	56.3	58.6	22.2	37.2	52.2
0.3	**73.7**	53.8	56.7	20.8	34.8	44.9
0.5	71.5	58.1	57.3	19.3	33.4	41.4

Table 7. Precision, recall, F1, average precision, and average recall for spatially conditioned image generation with our method, with different values for the threshold T. Metrics are averaged across 100 random samples from the MSCOCO [7] validation set and four random seeds. Average precision and average recall are calculated using DETR [4]. * The value used for our method

Fig. 8. Examples of spatially conditioned images generated with the similarity-based baselines and with our explainability-based method.

Fig. 9. Examples of spatially conditioned images generated with the similarity-based baselines and with our explainability-based method.

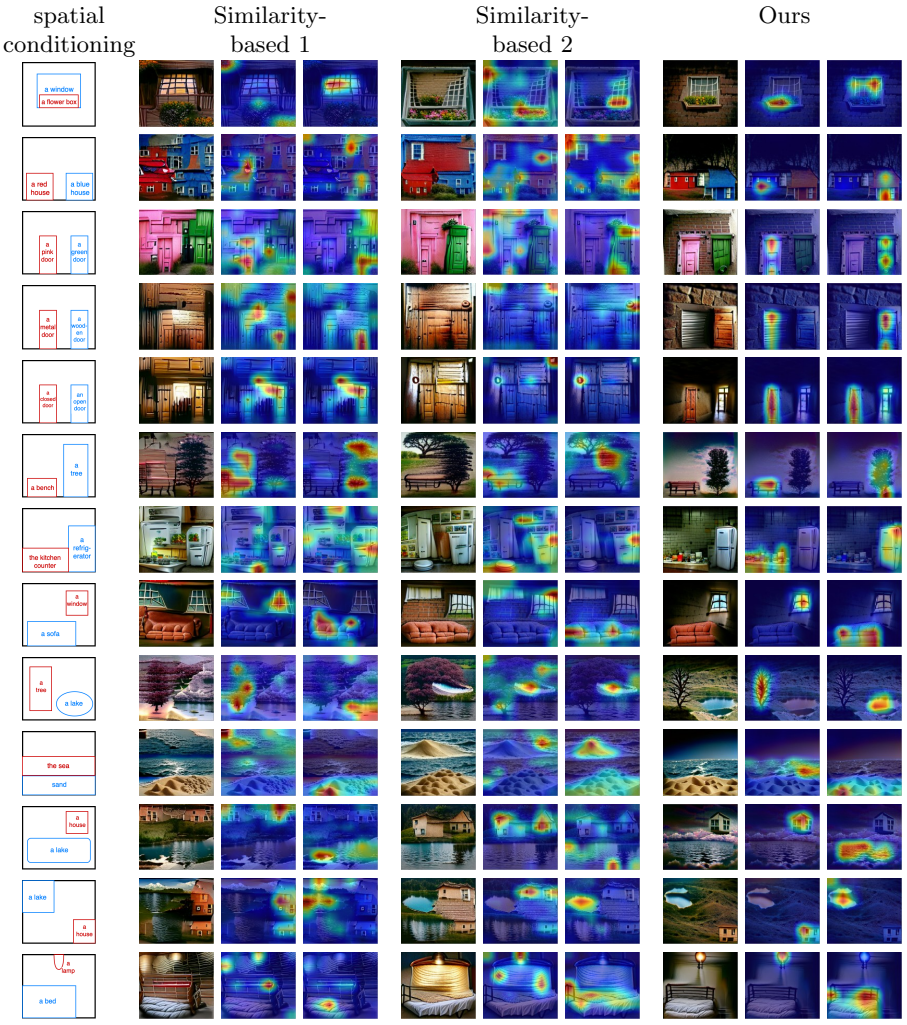

Fig. 10. Examples of spatially conditioned images generated with the similarity-based baselines and with our explainability-based method, along with the relevance maps produced for the images and their matching text prompts. The leftmost column presents the spatial conditioning provided to each method, each of the bounding boxes (red, blue) serves as a mask for the generation process. For each method, the left image is the generated results, the middle image is the explainability map for the object conditioned with the red bounding box, and the right image is the explainability map for the object conditioned with the blue bounding box. As can be seen, the relevancy maps correspond well to the objects for our methods, while the baselines suffer from significant artifacts.

λ_{expl}	Precision	Recall	F1	AP	AR	$AP_{0.5}$
$\frac{0.1}{\sqrt{r(m)}}$	70.5	62,4	61.3	22.1	37.0	48.2
$\frac{0.15}{\sqrt{r(m)}}$*	**71.7**	63.4	62.6	**26.2**	**40.0**	**56.5**
$\frac{0.2}{\sqrt{r(m)}}$	**71.7**	**64.1**	**62.7**	24.4	38.5	51.3

Table 8. Precision, recall, F1, average precision, and average recall for spatially conditioned image generation with our method, with different values for λ_{expl}. $r(m)$ denotes the ratio between the area of the mask and the area of the entire image. Metrics are averaged across 100 random samples from the MSCOCO [7] validation set and four random seeds. Average precision and average recall are calculated using DETR [4]. * The value used for our method

References

1. Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., Vollgraf, R.: Flair: An easy-to-use framework for state-of-the-art nlp. In: NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations). pp. 54–59 (2019)
2. Akbik, A., Blythe, D., Vollgraf, R.: Contextual string embeddings for sequence labeling. In: COLING 2018, 27th International Conference on Computational Linguistics. pp. 1638–1649 (2018)
3. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=B1xsqj09Fm
4. 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)
5. 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)
6. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf
7. 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)
8. 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)
9. 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)
10. 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)
11. Zhao, S., Liu, Z., Lin, J., Zhu, J.Y., Han, S.: Differentiable augmentation for data-efficient gan training. ArXiv **abs/2006.10738** (2020)
12. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. arXiv preprint arXiv:2109.01134 (2021)