Search-based Negative Prompt Optimisation for Text-to-Image Generation

Guillermo Iglesias[1][0000−0001−8733−7148], Mar Zamorano[2,3][0000−0002−8872−4876], and Federica Sarro[2][2222−−3333−4444−5555]

[1] Universidad Politécnica de Madrid, Madrid, Spain
guillermo.iglesias@upm.es
[2] University College London, London, United Kingdom
{maria.lopez.20,f.sarro}@ucl.ac.uk
[3] Universidad San Jorge, Zaragoza, Spain
mzamorano@usj.es

Abstract. Text-to-image generative models are machine learning models that take a description written in natural language as input and generate images matching this description. As with other types of generative models, text-to-image ones tend not to be precise due to various reasons, such as hallucinations or randomness, and are influenced by the input description (a.k.a. user's prompt). Therefore, their use might lead to images that do not fully meet user's expectations. Prompt engineering (i.e., the process of structuring text that can be interpreted and understood by a generative model) poses a significant challenge, demanding a considerable amount of manual effort to ensure high-quality image generation. In this work, we explore the use of a local search guided by sentence similarity to optimize text-to-image generation via negative prompts. Our results suggest that by using our approach, it is possible to improve the generation process, thus obtaining more accurate images with no additional human effort.

Keywords: Prompt engineering · Negative prompt · Sentence similarity · Image Generation · Generative AI · Local Search

1 Introduction

Generative AI (GenAI) has emerged as a significant revolution in the past decade [29] in many fields, including Arts [42]. The combination of Large Language Models (LLMs) with generative models has greatly improved the quality and possibilities of automated content generation [5]. These systems are trained using extensive datasets accompanied by specific descriptions, known as prompts, which play a crucial role in achieving the desired outcome [47]. However, the construction of these prompts can affect the results, sometimes leading to unexpected outcomes or errors [27].

The challenge of generating images exemplifies this issue. Generative models like DALL-E 2 [33] or Stable Diffusion [36] suggest modifying the prompt to ensure that the generated images align more closely with the desired expectations.

Prompt engineering consists of designing and modifying the user's prompts to improve the generation process, reducing hallucinations (i.e., discordance between prompt and output), and improving the overall quality of the output [7]. Along with the prompt, users may include terms to steer the diffusion toward the images associated with it through positive prompt or terms to steer the diffusion away from it through negative prompt. However, most of the time this process is performed by the user based on a tedious manual trial and error [32].

In this paper, we propose the use of local search to automatically search for an optimal prompt guided by automatic captioning and sentence similarity measurement, which eliminates the need for the end-user to manually optimize the prompt and check for the quality of the corresponding image at each step.

Our approach relies on the following intuition: given a prompt and the corresponding image generated by a given text-to-image tool, we expect the image to be of a good quality (i.e., it is an accurate visual representation of the textual prompt), if when using the image as input to a captioning tool we are able to generate a caption, which is very similar to the input prompt. In this work, we leverage this intuition to iteratively search for optimal prompts (in particular, negative prompts) guided by a fitness function based on automated caption and text-similarity.

Specifically, our approach starts from an initial solution represented by a user's prompt and corresponding image, and uses Stable Diffusion [36] to generate five additional images based on this prompt. BLIP [24] is then used to automatically generate a caption for each image, and compute the similarity between each caption and the user's prompt (i.e., $initialSolutionSimilarity$). Then, at each iteration the approach generates a set of neighbour solutions by removing one word at a time from each of the five captions (i.e., neighbours generation) and measuring their similarity with the user's prompt $neighbourSimilarity$ (i.e., fitness evaluation). If $neighbourSimilarity$ is higher than $initialSolutionSimilarity$, this suggests that we have found a term that may make the user's prompt worse, and therefore we add this term to the negative prompt. The concept of negative prompt, emerging from conditional generation models like Stable Diffusion, allows users to specify what to exclude from the generated images. The search continues until a maximum number of iterations is reached.

To evaluate our approach we use 17 pairs of prompts-images publicly available from the DiffusionDB [49] dataset. For each of these 17 pairs, we run our proposed approach and compared the text-similarity of the description of both the worst and best generated solutions against the original one (from now onwards referred to as baseline). In addition, we perform a human-study where participants were given the user's prompt and asked to rank the images based on their the quality.

Our results show that our approach improves the caption similarity obtained by 76.75%. Also, the human evaluation suggest that caption similarity is related to the human-perceived quality of the image: in fact, their assessment on average ranks as best the image with the highest similarity score (1.77 ± 0.71). This

indicates that our proposed fitness function based on automatic captioning is a good proxy for human-perceived image quality.

Moreover, both the worst and best solutions found by our approach show a higher similarity and hence better image-quality than the prompt-image baseline. Therefore, the proposed search-based prompt engineering guided by automatic captioning and text-similarity measurement has made it possible to optimise the quality of text-to-image generation, without the need of visually evaluating the images generated during the search or other human effort.

In summary, the contributions of this work are as follows:

1. we propose the first use of search for optimising the use of negative prompts in text-to-image generation; to this end, we devise a fitness function based on automatic captioning and text-similarity measurement to automatically evaluate the quality of images generated by text-to-image;
2. we realise our idea by implementing a local search that uses Stable Diffusion as an image generation tool and BLIP as a captioning tool, as these are popular and widely used publicly available tools;
3. we carry out an experimental validation of our proposal, and provide empirical results of its feasibility and effectiveness.

2 Related Work

Large Language Models for Text-to-Image: Large Language Models (LLMs) have become a hot research topic in recent years. Due to their nature they have been used in several areas such as content generation (i.e. text writing) [2], conversational agents (chatbots) [51], software development (code suggestions) [8], or translation [23]. One of the uses of LLMs is the generation of images, and more precisely, several LLMs and AI tools specialize in generating images from text descriptions. This means that given a text called *prompt*, the model will provide one or more unique images according to the text provided. Apart from the given text, more parameters can be modified in the model to obtain the desired images.

Some of the prominent ones are DALL-E [33], Stable Diffusion [1], MidJourney [19], Imagen [20], or Craiyon [10]. Each of these models has its strengths and specific use cases. When choosing a model for text-to-image generation, one needs to consider the individual characteristics of each model and the output desired. Factors to take into account are the quality and style of the images generated (e.g., photorealistic vs. artistic), the resolution and detail required, the availability and ease of use (including open-source options), and computational resources required to run the model.

Prompt Optimisation: Prompt optimization refers to the techniques used to improve the input text that the models need to generate content. There are recommendations when preparing text-to-image prompts such as being specific and detailed, using clear and concise language, including artistic and style descriptions, or using negative prompts [27]. Optimizing prompts for text-to-image

generation is crucial for obtaining high-quality, accurate, and relevant images. Adjusting the prompt is necessary to improve the model's comprehension of our intentions and enhance the quality of results, as noted by Reynolds and McDonell [35] and Zhou et al. [52]. This issue is particularly challenging in text-to-image models due to the limited capacity of their text encoders, such as the CLIP text encoder in Stable Diffusion [36].

Manual prompt engineering serves as a natural approach to prompt optimization. Despite the effectiveness of manually creating prompts, the process requires time and expertise [43], and may not always yield the best results [21]. Particularly when working with text-to-image models, users must meticulously select and compose sentences to achieve a specific visual style [27, 31]. Consequently, various methods focus on automatically searching for prompts through mining [21], paraphrasing [18], and text generation [15, 17]. In this paper, we propose for the first time the use of search to generate optimal negative prompts.

Image GenAI: Over the last decade, several GenAI systems have been proposed to carry out image generation tasks [22, 26]. The advancements in this field have led to gradual modifications or completely new structures, which include generator regularization [28], incorporation of additional memory gates [53], implementation of dynamic thresholding [37], and the fusion of neural architecture search with Generative Adversarial Networks [25]. While there is previous work that uses parameter tuning and/or prompt engineering to improve image generation [3, 12, 16], to the best of our knowledge our work is the first to propose a search-based approach to evolve negative prompts, coupled with the tuning of the guidance scale attribute of the text-to-image model.

Text-to-Image: The field of text-to-image generation has seen significant advancements with the development of models capable of transforming textual descriptions into corresponding visual content. A major area of research has focused on improving the input for generative models through effective prompt engineering. Recent studies have emphasized the importance of interactive prompt engineering, allowing users to iteratively refine their inputs to achieve more accurate and aesthetically pleasing images [13, 48]. Researchers have also explored the design principles behind effective prompt engineering for text-to-image models [27, 32]. In line with our work, some works put the focus on negative prompt engineering, such as Feng et al. [13] who used negative prompts based on iterative user feedback and Ding et al. [11] who used negative prompts based on the visual content of the input image, and Dong et al. [12] who used a learning strategy of positive-negative prompt-tuning.

3 Proposed Approach

Our approach aims to enhance image quality by incorporating appropriate terms in the negative part of the input prompt for text-to-image GenAI systems. Through a search-based approach, we look for suitable terms and iteratively build an optimal prompt that makes it possible to automatically produce higher quality and more precise images.

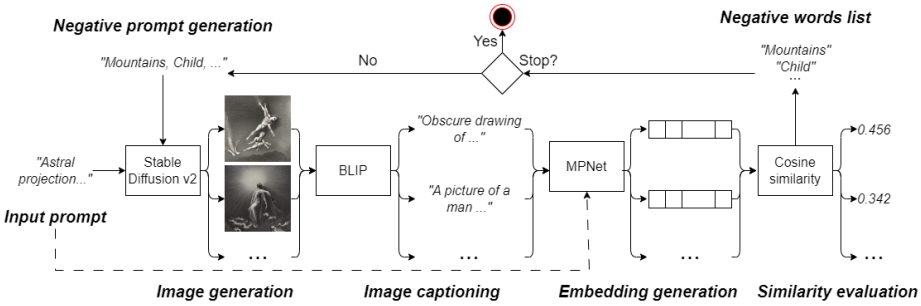

Fig. 1: Our search-based approach to optimal text-to-image generation using negative prompts.

Figure 1 shows an overview of our proposed approach. Given a prompt, a GenAI model generates a set of images (*Image generation*). This set of images is then given as input to another generative model capable of producing a text that describes each image, known as caption (*Image captioning*). From each newly generated caption, we want to extract those words that are misaligned with the input prompt, i.e., we want to compute their similarity. To do so, we convert the input prompt (indicated by the doted arrow) and each caption into an embedding, which converts the text into a representative vector of the words (*Embedding generation*). Using these embeddings, it is possible to calculate the similarity between texts using the cosine similarity (*Similarity evaluation*), thus providing feedback on the quality of the images. The more similar the captions, the higher the image quality. In addition, we use cosine similarity to calculate the impact of each term inside the vector (more detail about the impact in Section 3.4), thus identifying those terms that are misaligned with the input prompt (*Negative words list*). Such terms are injected into the negative prompt that complements the input prompt for the next iteration of our search-based approach. Once the maximum number of iterations is reached, the output is an optimised prompt which allows us to obtain an image with a caption that has the highest cosine similarity with the input prompt.

The following subsections provide more detail of the different steps of our approach: image generation, image captioning, embedding generation, and similarity evaluation.

3.1 Image Generation

The first step of our local search approach is the generation of a batch of N images given an input prompt. For each iteration of the search, our approach generates a new batch of images. To generate the images we use the Stable Diffusion v2 [36] model. Stable Diffusion v2 is a state-of-the-art generative model designed for creating high-quality images. It builds on the principles of diffusion models and has become popular for its ability to generate detailed, realistic images from text prompts.

Our approach provides a text prompt as required for the Stable Diffusion v2 model, and, as an additional input, it also provides a negative prompt and modifies the parameter guidance scale. The negative prompt is generally used to indicate what elements not to include in the image, while the guidance scale attribute weights the strength of the impact of the prompt on the generated image, the greater the value, the more closely the image follows the specified text input; however, higher values also result in less variety in the images and reduced quality.

Stable Diffusion v2 parameters are set as default: noise vector is defined as random, iteration steps are set to 50, latent space dimensionality as 512 and seed value as random. The rest of the parameters will be used in our optimization approach.

The input prompt remains constant during the search process. The negative prompt may vary on every iteration adding the new negative terms found at each iteration. To use the negative terms list to feed the negative prompt, a unique sentence with all the words must be generated. This sentence is constructed by concatenating each word in a unique sentence, separating each term with a comma. Thus, it is possible to combine all the terms in an unordered sentence that can be used to feed the network, e.g. with the set of negative terms *"tree"*, *"apple"* and *"dog"* the negative prompt would be *"tree, apple, dog"*.

On the other hand, our approach modifies the guidance scale attribute of Stable Diffusion v2 based on the average (i.e., mean) fitness of each iteration. This helps the process to perform a global search when the fitness is low and a local search when it is high. Its value varies within the range of [7, 13], with the fitness value ranging between [0.2, 0.6], clipping its value for lower or higher fitness. We determined these specific values due to the performance presented and taking into account that the values are within the normal values of the guidance scale (1.0 to 20.0) [49].

3.2 Image Captioning

The second step of our approach focuses on obtaining a text from which we can extract negative words to include in the negative prompt. To do so, we generate captions for the images generated in the previous step (Image generation). A caption from an image is a brief description or explanation that provides context about the image.

We use BLIP [24] for captioning generation. BLIP (Bootstrapping Language-Image Pre-training) is an advanced AI framework designed for vision-language tasks, such as image captioning, visual question answering, and image-text retrieval. BLIP aims to create more effective and generalizable models by leveraging a combination of supervised learning on labelled data and self-supervised learning on unlabelled data. BLIP provides a pre-trained model that only requires passing an image as input to obtain a caption as output.

3.3 Embedding Generation

From the captions generated in the previous step, one could manually analyse the input prompt and the caption to identify words that do not correspond to the input prompt. However, doing this process manually would consume great effort and time. Thus, the third step of our approach makes use of embedding to automate the comparison between the input prompt and each of the generated captions our approach.

Embedding refers to the process of mapping data from one representation to another, often in a lower-dimensional space while preserving certain properties of the original data. It is widely used in various fields, such as natural language processing (NLP), computer vision, and machine learning.

For each caption, an embedding is generated to extract its information. The embedding is a one-dimensional feature vector that represents the content of the text of the caption. We use MPNET (Multi-modal Pre-trained Networks) [45] to generate the embedding from text sources. MPNET is a model that is capable of generating a text embedding of 768 dimensions. Each embedding stores information about the semantic content of the text, making it possible to easily compare the contents of different text sources.

3.4 Similarity Evaluation

Finally, we use the embedding generated in the previous step with two purposes: first, to calculate the captions' similarity, and second, to identify the words that will be added to the negative prompt.

To calculate the similarity of captions, we use cosine similarity measures, following well-established practice in the literature[30, 14, 50]. Cosine similarity measures how similar two vectors are. In the context of text comparison, when using the proper trained encoder, the cosine similarity provides a value comparing how similar are the texts under comparison. The cosine similarity between two vectors is calculated as the cosine of the angle between them, which is also the dot product of the vectors divided by the product of their magnitudes (see equation 1).

$$\cos(A, B) = \frac{A \cdot B}{\|A\|\|B\|} \tag{1}$$

The similarity value between the input prompt and an image caption will measure how similar the content of the generated image is with respect to the asked prompt.

On the other hand, to identify the words that mislead image generation we evaluate each of the words in each caption. Thus, we evaluate one by one the impact of each word from each caption.

To calculate the impact of each word we calculate the cosine similarity between a caption without that word and the input prompt. If the similarity of the caption improves without the inclusion of a particular word, this word is a good candidate word to add to the negative prompt term (and we call this word

a negative term). We determine that a term is considered negative if the cosine similarity does not improve by more than 30%, because we observed that with a lower percentage the algorithm could not identify negative terms.

From the complete list of terms, we do not use English stop words, using the NLTK python package [4], a suite of libraries and programs for symbolic and statistical natural language processing.

4 Experimental Design

This section describes our research questions, dataset, human evaluation, models and experimental settings.

4.1 Research Questions

The research questions are formulated as follows:

RQ_1: *To what extent does our approach improve text-to-image generation?*

To answer $\mathbf{RQ}_1$ we compare the final similarity values achieved by the best and worst solutions generated by our approach with the original image. Specifically, we generate a caption for the original images from DiffusionDB, and following the same procedure as for our generated images we obtained the similarity between this caption and the input prompt ($baselineSimilarity$). Then, we compare $baselineSimilarity$ with the the similarity values achieved by our generated images, and assess which one is the highest.

RQ_2: *To what extent the use of similarity is effective to guide the search?*

To answer $\mathbf{RQ}_2$ we analyse the similarity value obtained by our approach for every prompt at the first iteration (i.e., when there are no negative words in the negative prompt) and the best similarity value obtained during the search process.

RQ_3: *To what extent is caption similarity a good proxy for human-perceived image quality?*

To answer $\mathbf{RQ}_3$ we ran a human evaluation to assess the quality of the original images and those generated by our approach. We compare the results of the human assessment against our obtained similarity to measure whether the images which are judged of high quality by the human participants, are also the images with high similarity.

In the following we explain in more details the methodology followed to answer these RQs.

4.2 Dataset

To assess the feasibility and effectiveness of our approach, we use 17 different prompts and their corresponding images extracted from the DifusionDB dataset [49]. This dataset contains 1.8 million unique prompts, from which we selected 17 based on the following exclusion criteria.

We excluded prompts not within [5, 25] words; prompts with commas and with the term 'and' to reduce complex and inconsistent prompts, such as *"intricate, 3 d, cybernetic hawk, style by caspar david friedrich and wayne barlowe and ted nasmith."* that add too many styles and expressions, making it difficult to generate and evaluate precisely the images. The resulting set of prompts eases the comparison and coherence of the texts, avoiding incoherent and subjective prompts.

These 17 pairs of prompt-image from DiffusionDB constitute the baseline of the methodology. Our process will generate new images using the same 17 prompts, and the resulting generated images will be evaluated against the baseline images.

4.3 Human Evaluation

We conducted an empirical evaluation of the quality of the images generated by our approach by disseminating an anonymous questionnaire. This questionnaire is meant to assess the quality of the generated images based on human evaluation. The questionnaire was disseminated publicly via social media such as Facebook, X and LinkedIn, and was open to everyone who is over 18 years old and has basic English knowledge. A total number of 53 participants answered the questionnaire, providing their feedback on the quality of the images.

The questionnaire contains 17 questions each corresponding to a distinct input prompt[4]. In particular, each question shows the participant the input prompt and three images: the image corresponding to the prompt from the DiffusionDB dataset (which we refer to as Base image), along with the best and worst image generated by our approach (which we refer to as best image and worst image). The images were presented in random orders and the users did not know how they were obtained. For each question, the participant is asked to rank the three images based on how well each of them reflects the given prompt.

Figure 2 shows two examples of prompts and the respective three images, as presented in the questionnaire. In addition, we include in Figure 2 the caption generated by our approach for each image.

4.4 Models

We used the Hugging Face implementation for the models, in particular: Stable Diffusion v2 from Stability AI [1], BLIP from Salesforce [38] and all-mpnet-base-v2 from SBERT [41].

We selected Stable Diffusion v2 to perform a fair comparison of the baseline DiffusionDB, as we inferred[5] that it is the model used to generate the images available in the DiffusionDB database. In addition, Stable Diffusion v2 is

[4] A copy of the questionnaire is available in the replication package.

[5] From authors of DiffusionDB words "We construct this dataset by collecting images shared on the Stable Diffusion public Discord server" [49], the date of the dataset (August 2022), and the release of Stability AI's Stable Diffusion v2 (June 2022); we deduce that the model used to obtain the images is Stable Diffusion v2.

Fig. 2: Examples of images generated in our experiment.

a publicly available widely used model that allows parameter setting and use of negative prompts.

BLIP is one of the most recent and advanced models for caption generation that combines vision and language transformers in an effective way [24]. In a recent work by Song and Song [44], BLIP demonstrates significant performance improvement in contrast to other previously established methodologies.

Finally, we selected the all-mpnet-base-v2 model from SBERT because it is one of the most widely used pre-trained models for tasks such as sentence similarity [34]. Due to its MPNet architecture [45], allows high-quality, context-sensitive sentence embeddings.

4.5 Experimental Settings

We use a simple search guided by text-similarity measurement to iterate over the image generation. The search continues until 50 iterations are reached for each input prompt. At each iteration, five different images (neighbours) are generated, captioned, and evaluated. We selected 50 iterations and five images as we observed it converges. Our final results using 50 iterations (see Section 5) show that the highest similarity (best solution) is achieved on average after 24.47 iterations, confirming that using 50 iterations was a sensible choice. Future studies could explore the use of fewer iterations. Table 1 shows in seconds the time needed for each step of our approach (note that the *Embedding Generation* step

Table 1: Average Computational Time (in seconds) taken by a single iteration of our approach across 5 images, and estimated average total time for 50 iterations.

Step	Single Iteration (s)	Total (s)
Image Generation	78.310	3915.5
Image Captioning	255.192	12759.6
Similarity Evaluation	0.047	2.35
Total	333.549	16677.45

Table 2: Summary of the results per prompt in terms of: similarity values (the higher the better) achieved by the baseline, the Best solution and Worst solution; number of negative words found after 50 iterations (Tot.) and the amount when reached the highest similarity (Best); iteration at which the best solution was found; and human ranking (the lower the better).

	Similarity			No. Iterations	No. Neg. Words	Human Evaluation		
ID	Baseline	Worst	Best	Best	Best (Tot.)	Baseline	Worst	Best
0	0.32	0.43	0.61	36	1 (1)	2.16	2.03	**1.79**
1	0.65	0.59	0.73	9	0 (0)	**1.92**	2.05	2.01
2	0.13	0.45	0.57	43	3 (4)	2.79	**1.56**	1.64
3	0.13	0.23	0.34	14	2 (17)	1.88	**1.69**	2.41
4	0.67	0.29	0.73	2	10 (13)	2.2	2.2	**1.58**
5	0.43	0.37	0.63	45	22 (22)	2.09	2.75	**1.15**
6	0.18	0.22	0.38	1	18 (43)	**1.94**	2.07	1.98
7	0.44	0.63	0.75	32	0 (1)	2.37	2.39	**1.22**
8	0.34	0.28	0.77	39	5 (7)	2.56	**1.62**	1.81
9	0.14	0.16	0.32	10	8 (40)	2.0	2.15	**1.84**
10	0.1	0.15	0.26	29	27 (38)	**1.39**	1.94	2.66
11	0.11	0.14	0.34	10	34 (67)	**1.66**	2.24	2.09
12	0.31	0.73	0.75	23	0 (3)	2.41	2.01	**1.56**
13	0.44	0.52	0.87	20	2 (2)	2.58	**1.56**	1.84
14	0.36	0.52	0.62	36	2 (2)	2.41	**1.69**	1.88
15	0.38	0.41	0.49	12	0 (8)	2.09	2.64	**1.26**
16	0.35	0.45	0.59	19	2 (4)	2.45	2.28	**1.26**
Mean	0.32	0.39	0.57	24.47	8 (16)	2.17	2.05	**1.76**

is included in the *Similarity Evaluation*). We can observe that the image captioning step (performed by using the BLIP tool) is the most time consuming, so future work may focus on parallelising this step or compare different tools to speed up the overall process.

All experiments were conducted on two 48 GB Nvidia Quadro RTX 8000 GPUs and an Intel Xeon Bronze 3206R CPU @ 1.90GHz.

The *similarity* is the principal evaluation measure in our evaluation. To calculate the *similarity* we use the cosine similarity as explained in Section 3.4. For all the comparisons we use the input prompt as the oracle. We compare against the oracle the captions of the generated images, and we also generate a caption from the original image to compare with.

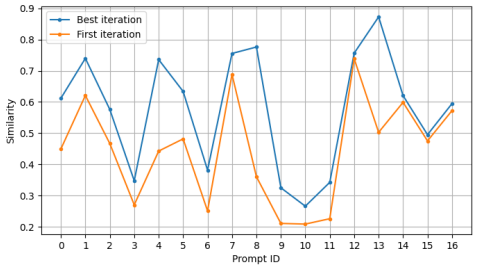

Fig. 3: **RQ**$_2$. Similarity values obtained by our approach at the first iteration and at the best iteration, for each prompt.

5 Results

To evaluate the performance of our local search algorithm we analyse the similarity of the caption of the images generated by our approach and the caption of the baseline images. The higher the similarity, the better. Table 2 shows the similarity values per prompt for the baseline and the best image and worst images generated by our search algorithm. We can observe that the mean similarity value achieved by the baseline is 0.32±0.17 are lower than the best images generated by our approach (which achieve a mean similarity value of 0.57±0.18). Additionally, even the worst images generated by our approach achieve a mean value higher than the baseline (i.e., 0.39±0.17).

Answer to RQ$_1$: The best images and worst images generated by our approach achieve a mean similarity value which is 76.75% and 19.5% higher than the baseline, respectively. This suggest that our approach generate better images according to the prompt in terms of similarity.

To analyse how the solutions evolve over the search, for each prompt, we report on the similarity values obtained within each iteration. Specifically, Figure 3 compares the similarity value achieved in the first iteration and the highest similarity achieved throughout the search. Depending on the image, the improvement in similarity ranges from 0.02 to 0.41, being 0.13 on average across the 17 prompts, from 0.44±0.16 on average on the first iteration to 0.57±0.18 on the highest similarity. In terms of iterations, on Table 2 we observed that on average 24.47 iterations were sufficient to achieve the highest similarity value (hence the best solution), which is about half of the total iterations performed (50 iterations). This observation is in line with the number of negative words, where the highest similarity required on average half (8±10.34) of the total number of negative words identified (16±20).

Answer to RQ$_2$: During the search, the similarity of the solutions improves on average by a factor of 29.54% since the first iteration. In addition, we observe how half of the iterations were sufficient to achieve the highest similarity.

Finally, to analyse whether text-similarity is a good proxy for human-perceived image quality, we present the results of the user study (see Table 2). From the

analysis of a total of 53 responses, we found that in 13 out of 17 cases the images generated by our approach (including best and worst images generated) were ranked higher than the baseline. The best image generated by our approach is ranked on average at 1.77±0.71, while the worst one is ranked at 2.06±0.67, and the image from DiffusionDB (i.e., our baseline benchmark) is ranked at 2.18±0.72. These results mean that our images generated (best and worst) were ranked on average first and second respectively more often than the baseline.

Analysing the four cases were the baseline was on average ranked first (IDs: 1, 6, 10, and 11), we noticed that for three of them the number of negative words found surpass by double or more the average of negative words. After a manual analysis of the images and prompts, we noticed that the prompts (namely '*astral projection by gustave dore*', '*the most amazing smore you have ever seen*', and '*steve jobs breaks the tablets of the law by gustave dore*') are more open to interpretation than the rest of the prompts in our dataset. This suggests that our approach is more effective when used with less ambiguous prompts (e.g., ID7: '*huge glitter bomb explosion above city*').

To further assess whether text-similarity is a good proxy for the human perception of image's quality we compute the Pearson's Correlation Coefficient [9].[6] We observe that there is a strong correlation (Pearson's coefficient=-0.52, p-value=0.031) between the similarity achieved by the best solution produced by our approach and the human evaluation scores. This means that the higher the similarity value, the lower (i.e.. the better) the quality judged by the human evaluators. In the other cases (i.e., baseline and worst solution produced) the results are not significant (p-values>0.26) and no correlation is observed.

Answer to RQ$_3$: The text-similarity is a good proxy for the human-perceived quality of the image, which suggests that text-similarity is a good fitness function for the problem at hand.

6 Threats to Validity and Discussion

To mitigate possible threats to **construction validity** we use a widely used and well-established measure such as cosine similarity as our fitness function and evaluation measure. In order to address the *lack of good descriptive statistics*, we present a table of the similarity results between our approach and the baseline as well as the comparison of the improvement between our first and our best image generated. We also present the mean and standard deviation of the empirical evaluation results. We tackled the *lack of a meaningful comparison baseline* by comparing our approach to 17 different cases selected from the public and widely used DiffusionDB dataset. Moreover, using instances from DiffusionDB, a publicly available dataset, which is widely used in the literature

[6] Statistical measure that evaluates the strength and direction of the relationship between two variables. A value near ±1 indicates a perfect correlation, ±0.50 - ±1 (strong correlation), ±0.30 - ±0.49 (moderate correlation), >+0.29 (weak correlation), and 0 (no relationship). A positive coefficient indicates that the variables are directly related, while a negative coefficient indicates inverse relationship.

enabled us to avoid the *lack of a clear object selection strategy*. To minimise possible **conclusion validity** threats, such as *not accounting for random variation*, we ran the approach 10 times. To minimise possible **internal validity** threats arising from the *lack of real problem instances* we use a public dataset made by real users. To reduce the threats arising from the choice of the models, we use well-known open-source models for captioning, embedding and image generation. However, to further minimise this threat it would be useful to compare the bias incurred by the models we used with other models such as GIT [46] for captioning or USE [6] for embedding. Moreover, we carefully describe the dataset, the *source code*, the approach *parameters* and the models used in this work, as well as provide a public replication package to facilitate replication, reproduction and extension of our work. We also carried out an anonymous user study in order to *validate* the quality of the generated images according to users' expectations. To mitigate possible **external validity** threats due to lack of *generalization*, we designed our approach to be generic for any input prompt.

7 Conclusions

We present a novel approach for optimizing GenAI image models. Our proposal relies on the combined use of prompt engineering and search-based optimisation to improve image generation. Specifically, we use sentence similarity to identify negative terms that we include as negative prompt through an iterative search process. Our results suggest that our contribution improves the adequacy of the generated images with respect to the input prompt, thus making them more precise and reducing hallucinations.

The proposed methodology relies heavily on the effectiveness of captioning and image generation models. Future work can analyse whether the use of different models may lead to strengthening and improving the robustness of the approach. In addition, it might be worth exploring whether other search-based approaches and fitness function configuration (e.g., guidance scale) could improve the results.

Furthermore, in this work, we focused only on image quality, however, other non-functional attributes can be optimised during the search, such as inference time and energy consumption [39, 40]. The results presented herein demonstrate a positive impact of using our proposed search-based process, however the magnitude of the improvement could be further studied with a larger human evaluation and more input prompts.

Replication Package The replication package is at:
https://github.com/guillermoih/Improving-GenAI-with-negative-prompts

Acknowledgments. Ethics approval has been obtained from the UCL CS ethics committee for the user study (UCL-CSREC-205-R). This work thanks the research group KNOwledge Discovery and Information Systems (KNODIS) for the compute infrastructure, and it has been partially supported by MINECO under the Project VARIATIVA (PID2021-128695OB-100), by the Gobierno de Aragón (Spain) (Research Group T61_23R), by the Excellence Network AI4Software (Red2022-134647-T).

References

1. AI, S.: Stable diffusion v2. Accessed 18/03/2024 (2022), https://huggingface.co/stabilityai/stable-diffusion-2
2. An, J., Ding, W., Lin, C.: Chatgpt. tackle the growing carbon footprint of generative AI **615**, 586 (2023)
3. Berger, H., Dakhama, A., Ding, Z., Even-Mendoza, K., Kelly, D., Menendez, H., Moussa, R., Sarro, F.: Stableyolo: Optimizing image generation for large language models. In: Search-Based Software Engineering: 15th International Symposium, SSBSE 2023, San Francisco, CA, USA, December 8, 2023, Proceedings. p. 133–139. Springer-Verlag, Berlin, Heidelberg (2023). https://doi.org/10.1007/978-3-031-48796-5_10
4. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc." (2009)
5. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
6. Cer, D., Yang, Y., Kong, S.y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al.: Universal sentence encoder for english. In: Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations. pp. 169–174 (2018)
7. Chen, B., Zhang, Z., Langrené, N., Zhu, S.: Unleashing the potential of prompt engineering in large language models: a comprehensive review. arXiv:2310.14735 (2023)
8. Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H.P.d.O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al.: Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374 (2021)
9. Cohen, I., Huang, Y., Chen, J., Benesty, J., Benesty, J., Chen, J., Huang, Y., Cohen, I.: Pearson correlation coefficient. Noise reduction in speech processing pp. 1–4 (2009)
10. Dayma, B.: Craiyon. Accessed 06/06/2024 (2024), https://www.craiyon.com
11. Ding, Z., Li, P., Yang, Q., Li, S.: Enhance image-to-image generation with llava prompt and negative prompt. arXiv preprint arXiv:2406.01956 (2024)
12. Dong, Z., Wei, P., Lin, L.: Dreamartist: Towards controllable one-shot text-to-image generation via positive-negative prompt-tuning. arXiv preprint arXiv:2211.11337 (2022)
13. Feng, Y., Wang, X., Wong, K.K., Wang, S., Lu, Y., Zhu, M., Wang, B., Chen, W.: Promptmagician: Interactive prompt engineering for text-to-image creation. IEEE Transactions on Visualization and Computer Graphics (2023)
14. Gao, A.K.: Vec2vec: A compact neural network approach for transforming text embeddings with high fidelity. arXiv preprint arXiv:2306.12689 (2023)
15. Gao, T., Fisch, A., Chen, D.: Making pre-trained language models better few-shot learners. arXiv preprint arXiv:2012.15723 (2020)
16. Gong, J., Li, S., d'Aloisio, G., Ding, Z., Ye, Y., Langdon, W.B., Sarro, F.: Greenstableyolo: Optimizing inference time and image quality of text-to-image generation. In: Jahangirova, G., Khomh, F. (eds.) Search-Based Software Engineering. pp. 70–76. Springer Nature Switzerland, Cham (2024)
17. Hao, Y., Chi, Z., Dong, L., Wei, F.: Optimizing prompts for text-to-image generation. Advances in Neural Information Processing Systems **36** (2024)

18. Haviv, A., Berant, J., Globerson, A.: Bertese: Learning to speak to bert. arXiv preprint arXiv:2103.05327 (2021)
19. Holz, D.: Midjouney. Accessed 06/06/2024 (2024), https://www.midjourney.com/home
20. Imagen: Imagen. Accessed 06/06/2024 (2024), https://imagen-ai.com/es/
21. Jiang, Z., Xu, F.F., Araki, J., Neubig, G.: How can we know what language models know? Transactions of the Association for Computational Linguistics **8**, 423–438 (2020)
22. Kang, M., Zhu, J.Y., Zhang, R., Park, J., Shechtman, E., Paris, S., Park, T.: Scaling up gans for text-to-image synthesis. In: Procs. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10124–10134 (2023)
23. Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., Chen, Z.: Gshard: Scaling giant models with conditional computation and automatic sharding. arXiv preprint arXiv:2006.16668 (2020)
24. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: International conference on machine learning. pp. 12888–12900. PMLR (2022)
25. Li, W., Wen, S., Shi, K., Yang, Y., Huang, T.: Neural architecture search with a lightweight transformer for text-to-image synthesis. IEEE Trans. on Network Science and Engineering **9**(3), 1567–1576 (2022)
26. Li, W., Zhang, P., Zhang, L., Huang, Q., He, X., Lyu, S., Gao, J.: Object-driven text-to-image synthesis via adversarial training. In: Procs. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12174–12182 (2019)
27. Liu, V., Chilton, L.B.: Design guidelines for prompt engineering text-to-image generative models. In: Procs. of the 2022 CHI Conference on Human Factors in Computing Systems. pp. 1–23 (2022)
28. Mao, Q., Lee, H.Y., Tseng, H.Y., Ma, S., Yang, M.H.: Mode seeking generative adversarial networks for diverse image synthesis. In: Procs. of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1429–1437 (2019)
29. Noy, S., Zhang, W.: Experimental evidence on the productivity effects of generative artificial intelligence. Science **381**(6654), 187–192 (2023)
30. Oikarinen, T., Weng, T.W.: Clip-dissect: Automatic description of neuron representations in deep vision networks. arXiv preprint arXiv:2204.10965 (2022)
31. Oppenlaender, J.: A taxonomy of prompt modifiers for text-to-image generation. Behaviour & Information Technology pp. 1–14 (2023)
32. Oppenlaender, J., Linder, R., Silvennoinen, J.: Prompting ai art: An investigation into the creative skill of prompt engineering. arXiv:2303.13534 (2023)
33. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv:2204.06125 **1**(2), 3 (2022)
34. Reimers, N.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
35. Reynolds, L., McDonell, K.: Prompt programming for large language models: Beyond the few-shot paradigm. In: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems. pp. 1–7 (2021)
36. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Procs. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10684–10695 (June 2022)
37. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in neural information processing systems **35**, 36479–36494 (2022)

38. Salesforce: Blip. Accessed 18/03/2024 (2024), https://huggingface.co/Salesforce/blip-image-captioning-large
39. Sarro, F.: Search-based software engineering in the era of modern software systems. In: Procs. of IEEE International Requirements Engineering Conference (2023)
40. Sarro, F.: Automated optimisation of modern software system properties. In: Procs. of the ACM/SPEC International Conference on Performance Engineering (2023)
41. SBERT: all-mpnet-base-v2. Accessed 18/03/2024 (2024), https://huggingface.co/sentence-transformers/all-mpnet-base-v2
42. Shahriar, S.: Gan computers generate arts? a survey on visual arts, music, and literary text generation using generative adversarial network. Displays **73**, 102237 (2022)
43. Shin, R., Lin, C.H., Thomson, S., Chen, C., Roy, S., Platanios, E.A., Pauls, A., Klein, D., Eisner, J., Van Durme, B.: Constrained language models yield few-shot semantic parsers. arXiv preprint arXiv:2104.08768 (2021)
44. Song, H., Song, Y.: Target research based on blip model. Academic Journal of Science and Technology **9**(1), 80–86 (2024)
45. Song, K., Tan, X., Qin, T., Lu, J., Liu, T.Y.: Mpnet: Masked and permuted pre-training for language understanding. Advances in neural information processing systems **33**, 16857–16867 (2020)
46. Wang, J., Yang, Z., Hu, X., Li, L., Lin, K., Gan, Z., Liu, Z., Liu, C., Wang, L.: Git: A generative image-to-text transformer for vision and language. arXiv preprint arXiv:2205.14100 (2022)
47. Wang, J., Liu, Z., Zhao, L., Wu, Z., Ma, et al.: Review of large vision models and visual prompt engineering. Meta-Radiology p. 100047 (2023)
48. Wang, Z., Huang, Y., Song, D., Ma, L., Zhang, T.: Promptcharm: Text-to-image generation through multi-modal prompting and refinement. In: Proceedings of the CHI Conference on Human Factors in Computing Systems. pp. 1–21 (2024)
49. Wang, Z.J., Montoya, E., Munechika, D., Yang, H., Hoover, B., Chau, D.H.: DiffusionDB: A large-scale prompt gallery dataset for text-to-image generative models. arXiv:2210.14896 [cs] (2022), https://arxiv.org/abs/2210.14896
50. Weerasinghe, E., Kotuwegedara, T., Amarasena, R., Jayasinghe, P., Manathunga, K.: Dynamic conversational chatbot for assessing primary students. In: International Conference on Artificial Intelligence in Education. pp. 444–448. Springer (2022)
51. Zhang, Y., Sun, S., Galley, M., Chen, Y.C., Brockett, C., Gao, X., Gao, J., Liu, J., Dolan, B.: Dialogpt: Large-scale generative pre-training for conversational response generation. arXiv preprint arXiv:1911.00536 (2019)
52. Zhou, Y., Muresanu, A.I., Han, Z., Paster, K., Pitis, S., Chan, H., Ba, J.: Large language models are human-level prompt engineers. arXiv preprint arXiv:2211.01910 (2022)
53. Zhu, M., Pan, P., Chen, W., Yang, Y.: Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In: Procs. of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5802–5810 (2019)