VQGAN-CLIP: Open Domain Image Generation and Editing with Natural Language Guidance

Katherine Crowson$^{\star 1}$, Stella Biderman^{*1,2}, Daniel Kornis[3], Dashiell Stander[1], Eric Hallahan[1], Louis Castricato[1,4], and Edward Raff[2]

[1] EleutherAI
[2] Booz Allen Hamilton
[3] AIDock
[4] Georgia Institute of Technology

Abstract Generating and editing images from open domain text prompts is a challenging task that heretofore has required expensive and specially trained models. We demonstrate a novel methodology for both tasks which is capable of producing images of high visual quality from text prompts of significant semantic complexity without any training by using a multimodal encoder to guide image generations. We demonstrate on a variety of tasks how using CLIP [37] to guide VQGAN [11] produces higher visual quality outputs than prior, less flexible approaches like minDALL-E [19], GLIDE [33] and Open-Edit [24], despite not being trained for the tasks presented. Our code is available in a public repository.

Keywords: generative adversarial networks; grounded language; image manipulation

1 Introduction

Using free-form text to generate or manipulate high-quality images is a challenging task, requiring a grounded learning between visual and textual representations. Manipulating images in an open domain context was first proposed by the seminal Open-Edit [24], which allowed text prompts to alter an image's content. This was done mostly with semantically simple transformations (e.g., turn a red apple green), and does not allow generation of images. Soon after DALL-E [38] and GLIDE [33] were developed, both of which can perform generation (and inpainting) from arbitrary text prompts, but do not themselves enable image manipulation.

In this work we propose the first a unified approach to semantic image generation and editing, leveraging a pretrained joint image-text encoder [37] to steer an image generative model [11]. Our methodology works by using the multimodal encoder to define a loss function evaluating the similarity of a (text, image) pair and backpropagating to the latent space of the image generator. We iteratively

* Co-first authors

update the candidate generation until it is sufficiently similar to the target text. The difference between using our technique for generation and editing is merely a matter of initializing the generator with a particular image (for editing) or with random noise (for generation).

A significant advantage of our methodology is the lack of additional training required. Only a pretrained image generator and a joint image-text encoder are necessary, while all three of Liu et al. [24], Ramesh et al. [38], and Nichol et al. [33] require training similar models from scratch. Additionally Ramesh et al. [38] and Nichol et al. [33] train generators from scratch.

We demonstrate several significant contributions, including:

1. High visual quality for both generation and manipulation of images.
2. High semantic fidelity between text and generation, especially when semantically unlikely content co-occurs.
3. Efficiency in that our method requires no additional training beyond the pretrained models, using only a small amount of optimization per inference.
4. The value of open development and research. This technique was developed in public and open collaboration has been integral to its rapid real-world success. Non-authors have already extended our approach to other modalities (e.g., replacing text for audio) and commercial applications.

The rest of our manuscript is organized as follows. In Section 2 we discuss how of how our methodology works, resulting in a simple and easy-to-apply approach for combing multiple modalities for generation or manipulation. The efficacy of VQGAN-CLIP in generating high quality and semantically relevant images is shown in Section 3, followed by superior manipulation ability in Section 4. The design choices of VQGAN-CLIP to obtain both high image quality and fast generation are validated by ablations in Appendix G, and Section 5 discusses resource usage and efficiency considerations. As our approach has been public since April 2021, we are able to show further validation by external groups in Section 6. This use includes extensions to other modalities, showing the flexibility of our approach, as well as commercial use of VQGAN-CLIP that demonstrate its success at handling open-domain prompts and images to a satisfying degree. Finally we conclude in Section 7.

2 Our Methodology

To demonstrate our method's effectiveness we apply it using VQGAN [11] and CLIP [37] as pre-trained models, and so refer to our approach as VQGAN-CLIP. We stress, however, that our approach is not specific to either model and that subsequent work has already shown success that builds on our work using other models [5, 27, 46, 12], and even in other modalities [18, 50].

We start with a text prompt and use a GAN to iteratively generate candidate images, at each step using CLIP to improve the image. We optimize the image by treating the squared spherical distance between the embedding of the candidate and the embedding of the text prompt as a loss function, and differentiating through CLIP with respect to the GAN's latent vector representation of the image, which we refer to as the "z-vector" following Oord, Vinyals, and Kavukcuoglu [35]. This process is outlined in Fig. 1

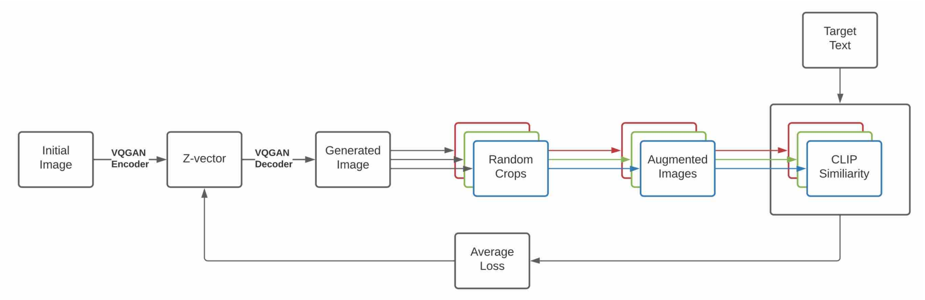


Figure 1: Diagram showing how augmentations are added to stabilize and improve the optimization. Multiple crops, each with different random augmentations, are applied to produce an average loss over a single source generation. This improves the results with respect to a single latent Z-vector.

To generate an image, the "initial image" contains random pixel values. The optimization process is repeated to alter the image, until the output image gradually improves such that it semantically matches the target text. We can also edit existing images by starting with the image-to-edit as the "initial image". The text prompt used to describe how we want the image to change is used identically to the text prompt for generating an image, and no changes to the architecture exist between generation and manipulation besides how the 'initial image" is selected.

We use Adam [20] to do the actual optimization, a learning rate of 0.15, $\beta = (0.9, 0.999)$, and run for 400 iterations for the experiments in this paper.

2.1 Discrete Latent Spaces for Images

Unlike the naturally discrete nature of text, the space of naturally occurring images is inherently continuous and not trivially discretized. Prior work by Oord, Vinyals, and Kavukcuoglu [35] borrows techniques from vector quantization (VQ) to represent a variety of modalities with discrete latent representations by building a codebook vocabulary with a finite set of learned embeddings. Given a codebook of vocabulary size K with embedding dimension n_k, $\mathcal{Z} = \{z_i\}_k^K \in \mathbb{R}^{n_k}$.

This is applied to images by constructing a convolutional autoencoder with encoder $\mathcal{E}$ and decoder $\mathcal{G}$. An input image $x \in I$ is first embedded with the encoder $z = E(x)$. We can then compute the vector quantized embedding x as

$$z_q = \underset{z_k \in \mathcal{Z}}{\operatorname{argmin}} \|z_{i,j} - z_k\|$$

which we can then multiply back through the vocabulary in order to perform reconstruction. We can then use a straight-through estimator on the quantization step in order to allow the CNN and codebook to be jointly trained end-to-end. We use the popular VQGAN [11] model for the experiments in this paper.

2.2 Contrastive Text-Image Models

To guide the generative model, we need a way to adjust the similarity of a candidate generation with the guidance text. To achieve this, we use CLIP, [37], a joint text-image encoder trained by using contrastive learning. We use CLIP to embed text prompts and candidate generated images independently and measure the cosine similarity between the embeddings. This similarity is then reframed as a loss that we can use gradient descent to minimize.

2.3 Augmentations

One challenge of using VQGAN-CLIP is that gradient updates from the CLIP loss are quite noisy if calculated on a single image. To overcome this we take the generated candidate image and modify it many times, producing a large number of augmented images. We take random crops of the candidate image and then apply further augmentations such as flipping, color jitter, noising, etc. [39] Most high level semantic features of an image are relatively invariant to these changes, so averaging the CLIP loss with respect to all of the augmented images reduces the variance of each update step. There is a risk that a random crop might dramatically change the semantic content of an image (e.g. by cropping out an important object), but we find that in practice this does not cause any issues.

For the results presented in this paper we used an augmentation pipeline consisting of: random horizontal flips, random affine projections, random perspective projections, random color jitter, and adding random Gaussian noise.

2.4 Regularizing the Latent Vector

When using an unconstrained VQGAN for image generation, we found that outputs tended to be unstructured. Adding augmentations helps with general coherence, but the final output will often still contain patches of unwanted textures. To solve this problem we apply a weighted L^2 regularization to the the z-vector.

This produces a regularized loss function given by the equation

$$Loss = L_{CLIP} + \alpha \cdot \frac{1}{N} \sum_{i=0}^{N} Z_i^2$$

where α is the regularization weight. This encourages parsimony in the representation, sending low information codes in VQGAN's codebook to zero. In practice we note that regularization appears to improve the coherence of the output and produces a better structured image. We decay the regularization term by 0.005 over the course of generation.

2.5 Additional Components

Our methodology is highly flexible and can be extended straightforwardly depending on the use-case and context due to the ease of integrating additional interventions on the intermediate steps of image generation. Researchers using our framework have introduced a number of additional components, ranging from using ensembles [6], to using Bézier curves for latent representations [13, 5], to using perturbations to make the results more robust to adversaries [25]. Although they aren't used in the main experiments of this paper, we wish to call attention to two in particular that we use frequently: "prompt addition" and masked image editing. We give an overview of both here, and provide additional experiments and information in Appendix H

Prompt Addition: We have found that our users are often interested in applying multiple text prompts at the same time. This can be achieved by computing the loss against multiple target texts simultaneously and adding the results. In Appendix H.1 we use this tool to explore the semantic cohesion of VQGAN-CLIP's generations.

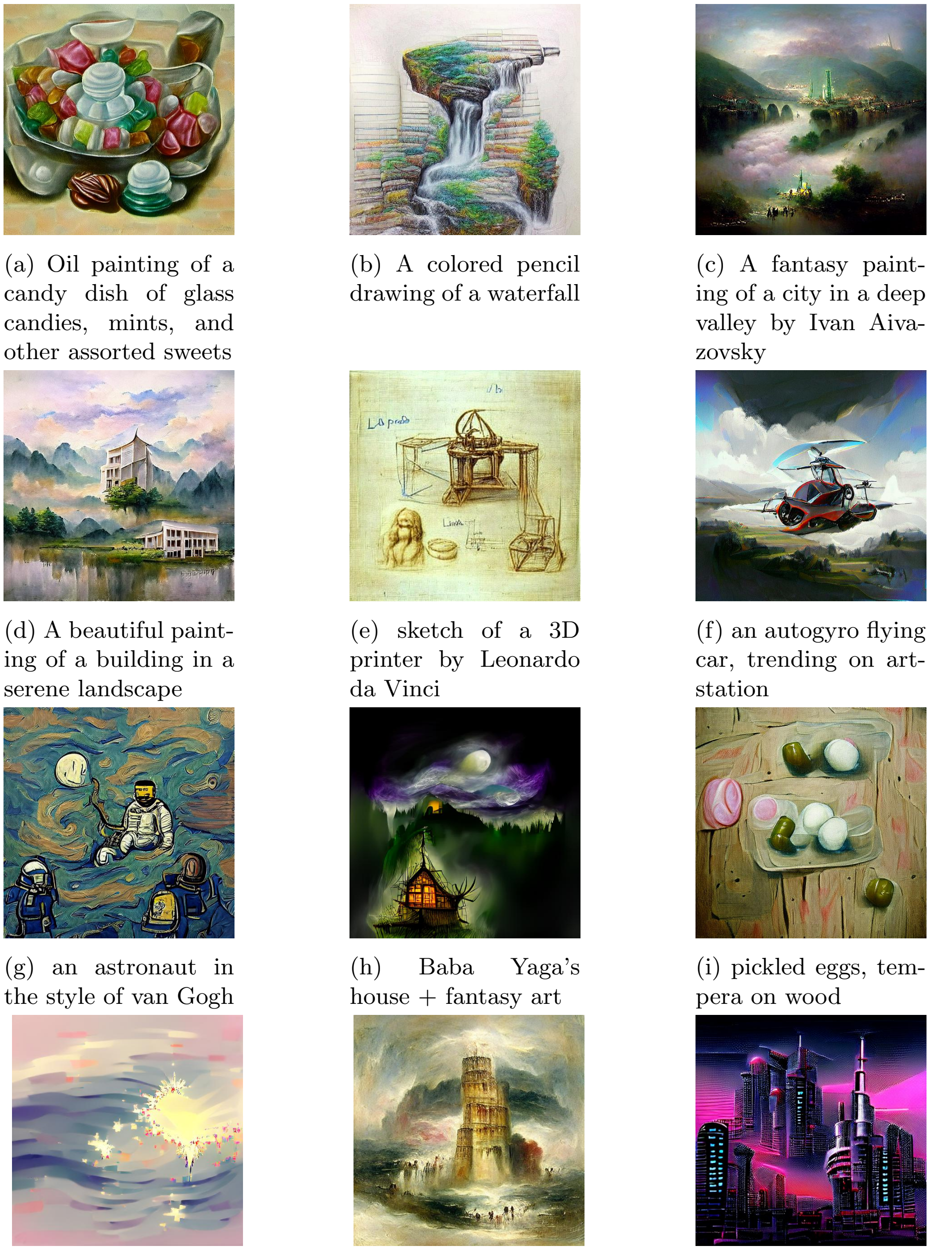
Masking: A common technique in image generation and editing is *masking*, where a portion of an image is identified ahead of time as being where a model should edit[5] VQGAN-CLIP is compatible with masking by zeroing out the gradients in parts of the latent vector that one wishes to not change. However VQGAN-CLIP can also leverage the semantic knowledge of CLIP to perform *self-masking* without any non-textual human input.

3 Semantic Image Generation

The primary application of our methodology is for generating images from text. In contrast to previous work on this topic [38, 33, 49], we do not perceive creating photo-realistic images or images that could convince a human that they are real photographs as our primary goal. Our focus is on producing images of high visual quality that are semantically meaningful in relation to a natural language prompt, which we demonstrate in this section. This in fact requires abandoning photo-realism when prompts may ask for artistic or explicitly unrealistic generations and edits. A loosely curated set of example generations is presented in Fig. 2.

As VQGAN-CLIP has been publicly available for almost a year, we have had the opportunity to observe people experimenting with and building off of VQGAN-CLIP in the wild. In Appendix D we show a sample of artwork created by people other than the authors of this paper are included to demonstrate the power and range of VQGAN-CLIP.

[5] In the context of image this is often referred to as "infilling," but we will use "masking" as a general term to refer to both.



(a) Oil painting of a candy dish of glass candies, mints, and other assorted sweets

(b) A colored pencil drawing of a waterfall

(c) A fantasy painting of a city in a deep valley by Ivan Aivazovsky

(d) A beautiful painting of a building in a serene landscape

(e) sketch of a 3D printer by Leonardo da Vinci

(f) an autogyro flying car, trending on artstation

(g) an astronaut in the style of van Gogh

(h) Baba Yaga's house + fantasy art

(i) pickled eggs, tempera on wood

(j) effervescent hope

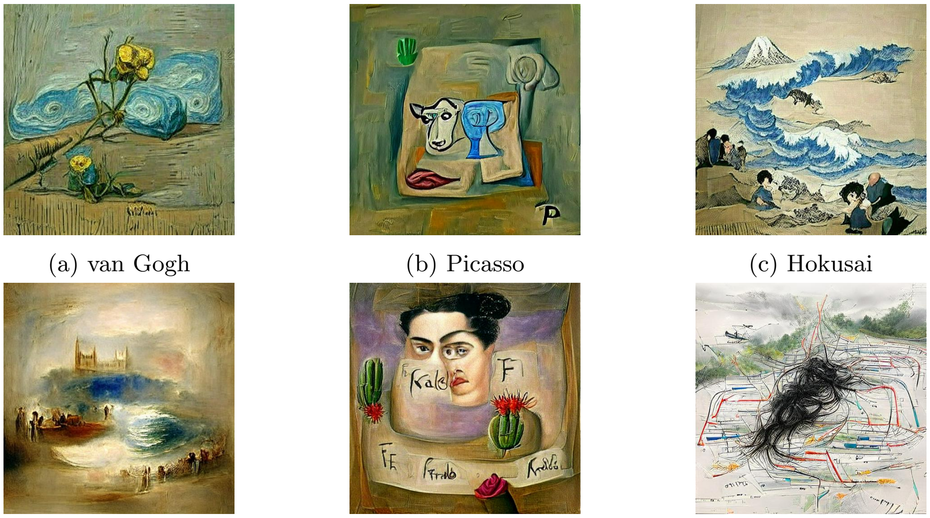
(k) the Tower of Babel by J.M.W. Turner

(l) a futuristic city in synthwave style

Figure 2: Example VQGAN-CLIP generations and their text prompts. Prompts selected to demonstrate a range of visual styles that VQGAN-CLIP is capable of producing including classical art (g, i), modern art (l), drawings (e), oils (a), and others not included due to space.

3.1 Artistic Impressions

We find that VQGAN-CLIP is able to evoke the artistic style of famous artists and major artistic styles from around the world. Fig. 2 features "an astronaut in the style of van Gogh" whose background evokes Starry Night and "the Tower of Babel by J. M. W. Turner" which draws on Turner's color palate and use of light. Another way this can be seen is by directly asking for "a painting by [name]" or "art by [name]." In Fig. 3 we present six images created this way drawing on artists from different regions, time periods, and artistic styles. While the images often are missing cohesion (most likely due to the vagueness of "a painting" as a prompt) they are each markedly reminiscent of the artist in question. While CLIP would obviously find these images visually similar to other works by the artist, we also find that non-CLIP-based image similarity approaches reliably identify these images as visually similar to work by the artists. To validate this we queried Google's Reverse Image Search using each generation in Fig. 3, and in every case a real painting by the target artist was the most similar image.



(a) van Gogh (b) Picasso (c) Hokusai

(d) Turner (e) Kahlo (f) Mehretu

Figure 3: Stylistic impressions of famous artists. Third party tools like Google's Reverse Image Search indicate that real paintings by the target artists are the most visually similar images in every case.

3.2 Comparisons to Other Approaches

The closest prior work in open domain generation of images comes from DALL-E [38] and GLIDE [33], which claim to train very large pretrained text-to-image models. DALL-E and GLIDE are purported to be 12 billion and 5 billion parameters, respectively, while VQGAN-CLIP together is 227 million. Unfortunately, we were not allowed to study the models purported in the respective papers by their authors. We instead use the state-of-the-art models using each methodology methodologies. This includes minDALL-E [19] (1.3 B parameters) and two versions of GLIDE (783 M parameters without CLIP and 941 M with) that OpenAI has released.

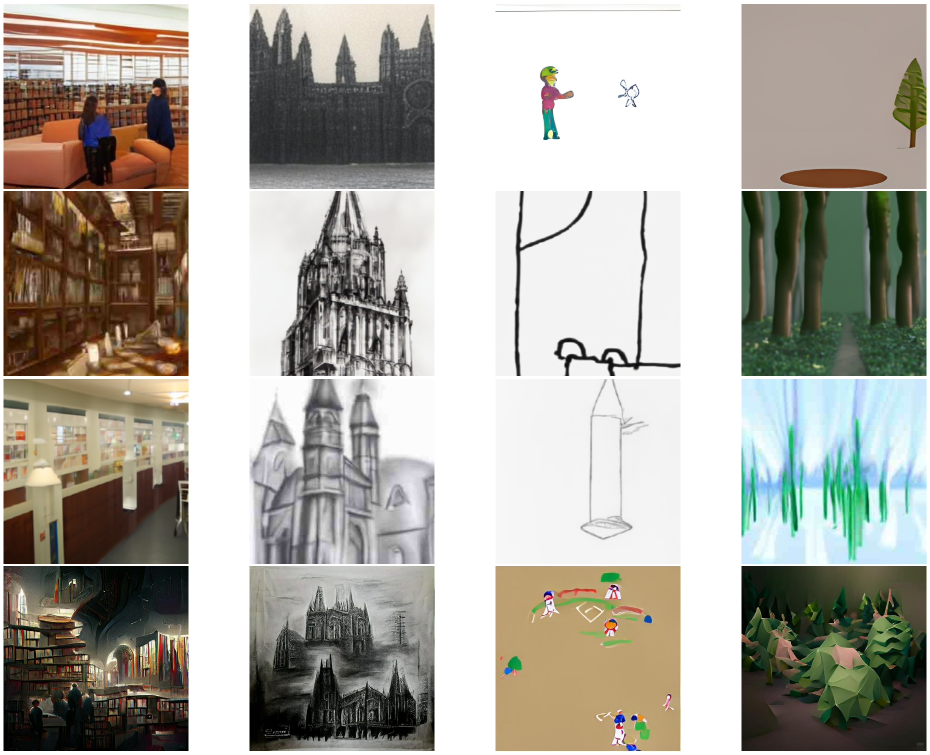
To evaluate our model, we recruited humans and asked them to rate the alignment of (text, image) pairs on a scale of 1 (low) to 5 (high). In particular, they were directed to rate higher quality images that do not match the prompt lower than lower quality images that do. Prompts were selected based on principles learned from our experience working with these models but without prior knowledge of how the models would behave on the particular prompts in question. All prompts and generated images can be found in Appendix D. To provide the maximal advantage to our competitors, minDALL-E and GLIDE examples are cherry-picked best-of-five, while VQGAN-CLIP examples are uncherry-picked (best-of-one). Table 1 shows the mean score per prompt for each model. We find that humans overwhelmingly view the generations using our technique as more aligned with the input text.

	A	B	C	D	E	F	G	H	I	J	K	L	**Mean**
minDALL-E	3.3	2.3	3.2	3.7	1.5	2.7	2.2	1.3	3.3	3.0	3.5	2.3	2.7
GLIDE (CF)	3.0	4.0	2.7	3.2	1.3	2.5	1.3	1.2	2.0	2.3	2.0	2.8	2.3
GLIDE (CLIP)	3.2	4.0	2.8	4.8	3.0	2.7	1.8	2.5	3.7	2.3	3.7	**5.0**	3.3
VQGAN-CLIP	**4.3**	**5.0**	**4.8**	**5.0**	**4.5**	**4.5**	**4.8**	**4.7**	**4.2**	**3.8**	**4.7**	4.7	**4.6**

Table 1: Mean human ratings of generations by each model considered on a score of 1 (worst) to 5 (best).

3.3 Qualitative Analysis

A sampling of representative results is shown in Fig. 4 for four different prompts using minDALL-E, two variants of GLIDE (filtered), and our VQGAN-CLIP. Further comparisons, including the prompts in Fig. 2, can be found in Appendix E. We find that the minDALL-E and the GLIDE (filtered) models are much more variable in the quality of their generations. While they are able to produce images that are clearly recognizable in response to the prompts “the universal library trending on artstation” and “a charcoal drawing of a cathedral,” their generations in response to “a child’s drawing of a baseball game” are largely unrecognizable and their responses to “a forest rendered in low poly” ignore the later half of the prompt. These latter cases demonstrate the low semantic relevance of prior methods’ output given the prompt.



(a) the universal library trending on artstation

(b) a charcoal drawing of a cathedral

(c) a child's drawing of a baseball game

(d) a forest rendered in low poly

Figure 4: Text based generations of images. Top to bottom: minDALL-E, GLIDE (CLIP-guided), GLIDE (CF-guided), and our VQGAN-CLIP.

The "child's drawing" case is of particular note here in that a child's drawing is expected to have lower visual clarity and lack of structure. That VQGAN-CLIP is able to correctly modulate its ability for fine details is thus of note to show that VQGAN-CLIP is not intrinsically biased toward producing fine details when inappropriate, and correctly identifies the appropriate context of multi-part prompts. Further evidence of this can be found in Figs. 9 to 12 where VQGAN-CLIP is able to produce generations for the prompts "A colored pencil drawing of a waterfall" and "sketch of a 3D printer by Leonardo da Vinci" that showcase the properties of the medium (visible strokes, the use of shading, the fact that the image is created on a piece of paper) while still producing a compelling visual image.

4 Semantic Image Editing

As far as we are aware, our framework is the first in the literature to be able to perform semantic image generation *and* semantic image editing. There are other examples in the literature of generative models that can perform style transfer [36], image inpainting [38, 33], and other types of image manipulation [34], but we note that each of these represent distinct tasks from open domain semantic image editing. By contrast, to adapt our generation methodology to image editing all that is required is to replace the randomly initialized starting image with the image we wish to edit.

4.1 Comparison to State-of-the-Art

For semantic image editing we compare to Open-Edit [24]. As far as we are aware, Open-Edit is the only published research on *open domain* semantic image editing other than our work. To avoid giving any accidental advantage to our methodology, we focus primarily on the domains presented as examples in Liu et al. [24] such as changing colors and textures. We use the default settings for their model and the same prompting structure as in their paper.

Color editing Here we prompt the model to change the dominant color palette without degrading the image quality or any of the finer details. The results can be seen in Fig. 5, where prior Open-Edit causes destructive transformations of the content of the image. In the second case the “Red bus” also shows a single desired target for manipulation that is respected by VQGAN-CLIP, but Open-Edit causes a change in coloration of the entire image.

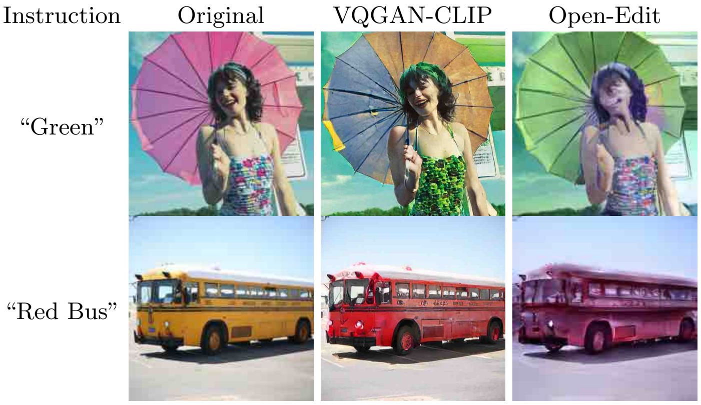


Figure 5: Examples of editing the color in an image. Original on the left, our VQGAN-CLIP in the middle, and Open-Edit on the right. VQGAN-CLIP better maintains original structure of the content while limiting unintended distortion.

Weather Modification Another use case that Liu et al. [24] highlight as a success of their model is weather modification, changing the overall weather conditions present in an image. Results on this task are shown in Fig. 6, where Open-Edit's reliance on edge maps to maintain structure show a limitation in editing ability. The needed alterations often change more of the image content that would violate the edge maps, preventing Open-Edit from being as successful in achieving the desired content change.

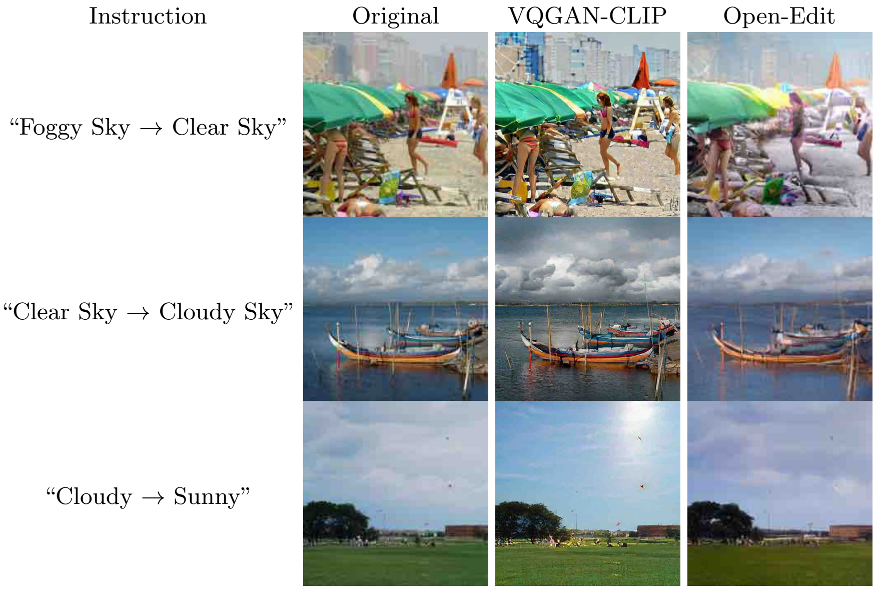


Figure 6: Weather alteration can required greater alteration of scene structure that Open-Edit is not able to perform, as shown in the “Cloudy → Sunny” example that needs to alter the sky in addition to brightness levels.

Misc We include extra miscellaneous examples to emphasize that this is open domain image editing and the performance is not limited to select types of transformations. These are shown in Fig. 7, and we note the “wooden” and “focused” examples demonstrate a task with less correlative semantics. This further requires a more robust grounding between modalities for success and the ability of our approach to better handle a breadth of possible inputs for open-domain prompts and images.

5 Resource Considerations

Our approach runs in $(935.2 \pm 20.4)\,\mathrm{s}$ on an NVIDIA Tesla K80 and $(229.5 \pm 26.2)\,\mathrm{s}$ on an NVIDIA GeForce RTX 2080 Ti (10 runs in each sample). This is

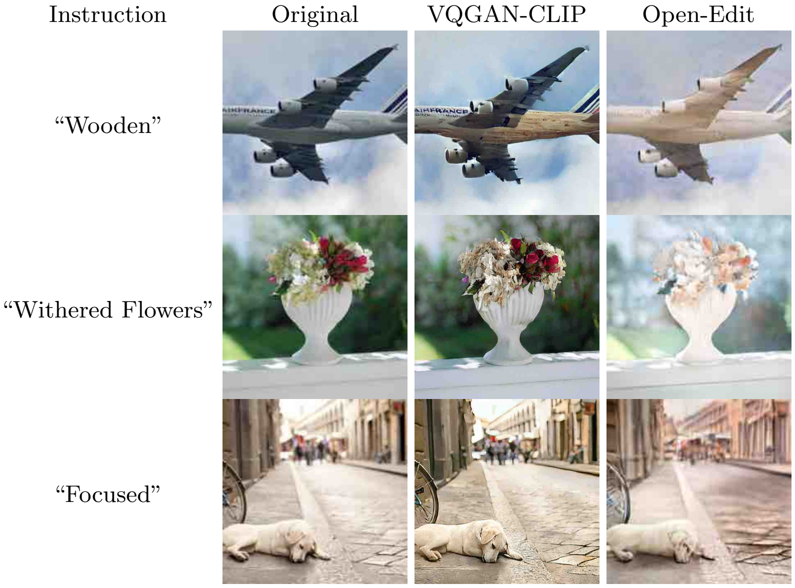


Figure 7: More challenging modifications that required greater linguistic grounding to visual content to achieve, again showing VQGAN-CLIP is better able to edit image content.

approximately three times slower than minDALL-E and ten times slower than GLIDE (filtered) in our testing. Although we would like to analyze the trade-offs involved with the fact that both models required extensive pretraining none of the papers we compare to report their training requirements in enough detail to analyze the trade-offs brought about by this difference. We encourage the authors to release more information about their models so that more complete analysis can be done.

5.1 Efficiency as a Value

One of the goals of this research is to increase the accessibility of AI image generative and editing tools. We have deliberately limited our approach to something that requires less than 11 GB of VRAM, so that it fits inside widely available commercial GPUs such as K80s. This GPU is particularly important from an accessibility standpoint, as it is the largest GPU that can be easily obtained using a free account on Google Colaboratory. The full generative process takes less than 3 minutes in a Google Colab notebook, making this a viable approach for anyone with access to the internet and a Google account.

Researchers with significantly more resources can obtain higher quality images using various augmentations left out of this paper, such as using an ensemble or additional auxiliary models to regularize the generations. While pushing the performance of our methodology to the maximum is a worthwhile endeavour, the fact that we can outperform the current state-of-the-art while running on freely available resources is something that we view as particularly worth highlighting. We leave determining the optimal framework with unbound resources to future work.

5.2 Runtime Analysis

DALL-E [38], GLIDE [33], and Open-Edit [24] all also incorporate image generators and joint text-image encoders into their architecture. Unlike our method however, they require computationally intensive training and finetuning. This invites the question of trade-offs between training and inference time. Unfortunately, none of the aforementioned papers report their training requirements in enough detail to estimate their training requirements. We are however able to estimate how long minDALL-E [19], the current state-of-the-art DALL-E model, takes to train at 504 V100-hours for the base model plus an additional 288 V100-hours to finetune on ImageNet [8]. Through private communication with the authors, we were able to learn that GLIDE (filtered) required 400 A100-days to train, which we approximate as $19,200$ V100-hours for ease of comparison.

Model	K80	P100	V100	Training
minDALL-E	$216.0s \pm 07.6s$	$60.0s \pm 5.5s$	$016.3s \pm 2.7s$	792 V100-hours
GLIDE (filtered)	$096.2s \pm 00.1s$	$19.2s \pm 0.3s$	$009.7s \pm 1.1s$	$19,200$ V100-hours
VQGAN-CLIP	$935.2s \pm 20.4s$	$654.3s \pm 10.1s$	$188.3s \pm 1.2s$	0 V100-hours

Table 2: Run-time of minDALL-E, GLIDE (filtered) and VQGAN-CLIP on a variety of GPUs. Each cell shows the mean and standard deviation of a 10-run sample. minDALL-E becomes cheaper than VQGAN-CLIP after 858 V100-hours have been expended while GLIDE (filtered) requires 20200 V100-hours.

On all hardwares evaluated, our model is substantially slower than both minDALL-E and GLIDE (filtered). However. In terms of trade-offs between training and inference on V100 GPUs, minDALL-E's total cost becomes cheaper than VQGAN-CLIP at $\approx 15,800$ generations, while GLIDE(filtered) requires $\approx 384,000$. In terms of compute expended, minDALL-E becomes cheaper than VQGAN-CLIP after 858 V100-hours while GLIDE (filtered) requires 20200 V100-hours. While cost and efficiency concerns depend significantly on individual contexts, the fact that GLIDE (filtered) only becomes as efficient as VQGAN-CLIP efficient after tens of thousands of dollars of compute have been expended substantially limits researchers' ability to experiment with and iterate on the methodology. The same applies to minDALL-E, albeit with a price tag in the thousands rather than tens of thousands.

6 Adoption of VQGAN-CLIP

A unique aspect of VQGAN-CLIP has been its public development over the past year, which has resulted in an active community of users and real-world impact within and beyond classical computer vision. Kwon and Ye [21], Frans, Soros, and Witkowski [13], Chen, Dumay, and Tang [5], Liu et al. [25], and Tian and Ha [46] create additional components (see Section 2.5) that they insert into our framework to improve performance in particular target domains, and Avrahami, Lischinski, and Fried [2] and Gu et al. [15] experiment with diffusion models in place of VQGAN. Several other researchers [27, 12, 33] evaluate their pretrained models by substituting them in for VQGAN or CLIP in our framework.

Beyond computer vision, Yang and Buehler [51] show that it is useful in the materials engineering design processes. Wu et al. [50] and Jang, Shin, and Kim [18] builds on our work by using the framework to perform sound-guided image generation. In the domain of affective computing and HCI, Galanos, Liapis, and Yannakakis [14] has further found VQGAN-CLIP able to elicit targeted emotions from viewers.

This last example helps explain the widespread commercial adoption of VQGAN-CLIP, with over a dozen commercial apps built to provide it as a service and over 500 NFTs produced using our method sold. A sampling of commercial websites using VQGAN-CLIP include NightCafe, Wombo Art, snowpixel.app, starryai.com, neuralblender.com, and hypnogram.xyz. Collectively, across these sites, VQGAN-CLIP has been used over 10 million times, showing the veracity of our approach to handle unstructured and diverse user content.

7 Conclusion

We have presented VQGAN-CLIP, a method of generating and manipulating images based on only human written text prompts. The quality of our model's generations have high visual fidelity and remain faithful to the textual prompt, outperforming prior approaches like DALL-E and GLIDE. The fidelity has been externally validated by commercial success and use by multiple companies. Compared to the only comparable approach to text based image editing, VQGAN-CLIP continues to produce higher quality visual images — especially when the textual prompt and image content have low semantic similarity.

Acknowledgements

We would like to acknowledge Ryan Murdock, who developed a very similar technique for combining VQGAN and CLIP simultaneously to us [30, 31] but did not release his approach. Ryan was invited to coauthor this paper and declined. Many of the experiments in this paper were made possible by the free public demo for VQGAN-CLIP in the EleutherAI Discord server developed by BoneAmputee and sponsored by CoreWeave.

References

1. Ali, S., and Parikh, D.: Telling Creative Stories Using Generative Visual Aids, (2021). arXiv: `2110.14810v1 [cs.HC]`
2. Avrahami, O., Lischinski, D., and Fried, O.: Blended Diffusion for Text-driven Editing of Natural Images, (2021). arXiv: `2111.14818v1 [cs.CV]`
3. Bau, D., Liu, S., Wang, T., Zhu, J.-Y., and Torralba, A.: Rewriting a deep generative model. In: European Conference on Computer Vision, pp. 351–369 (2020)
4. Black, S., Biderman, S., Hallahan, E., Anthony, Q., Gao, L., Golding, L., He, H., Leahy, C., McDonell, K., Phang, J., Pieler, M., Prashanth, U.S., Purohit, S., Reynolds, L., Tow, J., Wang, B., and Weinbach, S.: GPT-NeoX-20B: An Open-Source Autoregressive Language Model. preprint (2022)
5. Chen, G., Dumay, A., and Tang, M.: diffvg+CLIP: Generating Painting Trajectories from Text. preprint (2021)
6. Couairon, G., Grechka, A., Verbeek, J., Schwenk, H., and Cord, M.: FlexIT: Towards Flexible Semantic Image Translation, (2022). arXiv: `2203.04705 [cs.CV]`
7. De Cao, N., Aziz, W., and Titov, I.: Editing Factual Knowledge in Language Models, (2021). arXiv: `2104.08164v2 [cs.CL]`
8. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255 (2009)
9. Dong, H., Yu, S., Wu, C., and Guo, Y.: Semantic Image Synthesis via Adversarial Learning. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 5706–5714 (2017)
10. Eichenberg, C., Black, S., Weinbach, S., Parcalabescu, L., and Frank, A.: MAGMA – Multimodal Augmentation of Generative Models through Adapter-based Finetuning, (2021). arXiv: `2112.05253v1 [cs.CV]`
11. Esser, P., Rombach, R., and Ommer, B.: Taming Transformers for High-Resolution Image Synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12873–12883 (2021)
12. Fei, N., Lu, Z., Gao, Y., Yang, G., Huo, Y., Wen, J., Lu, H., Song, R., Gao, X., Xiang, T., Sun, H., and Wen, J.-R.: WenLan 2.0: Make AI Imagine via a Multimodal Foundation Model, (2021). arXiv: `2110.14378v1 [cs.AI]`
13. Frans, K., Soros, L., and Witkowski, O.: CLIPDraw: Exploring Text-to-Drawing Synthesis through Language-Image Encoders, (2021). arXiv: `2106.14843v1 [cs.CV]`
14. Galanos, T., Liapis, A., and Yannakakis, G.N.: AffectGAN: Affect-Based Generative Art Driven by Semantics. In: 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW) (2021)
15. Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., and Guo, B.: Vector Quantized Diffusion Model for Text-to-Image Synthesis, (2021). arXiv: `2111.14822v3 [cs.CV]`
16. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S.: Parameter-efficient transfer learning for NLP. In: International Conference on Machine Learning, pp. 2790–2799 (2019)
17. Hu, X., Yu, P., Knight, K., Ji, H., Li, B., and Shi, H.: MUSE: Textual Attributes Guided Portrait Painting Generation. In: 2021 IEEE 4th International Conference on Multimedia Information Processing and Retrieval (MIPR), pp. 386–392 (2021)
18. Jang, J., Shin, S., and Kim, Y.: Music2Video: Automatic Generation of Music Video with fusion of audio and text, (2022). arXiv: `2201.03809v1 [cs.SD]`
19. MISC

20. Kingma, D.P., and Ba, J.: Adam: A Method for Stochastic Optimization, (2014). arXiv: 1412.6980v9 [cs.LG]
21. Kwon, G., and Ye, J.C.: CLIPstyler: Image Style Transfer with a Single Text Condition, (2021). arXiv: 2112.00374v2 [cs.CV]
22. Lester, B., Al-Rfou, R., and Constant, N.: The Power of Scale for Parameter-Efficient Prompt Tuning, (2021). arXiv: 2104.08691v2 [cs.CL]
23. Li, B., Qi, X., Lukasiewicz, T., and Torr, P.H.: Manigan: Text-guided image manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7880–7889 (2020)
24. Liu, X., Lin, Z., Zhang, J., Zhao, H., Tran, Q., Wang, X., and Li, H.: Open-edit: Open-domain image manipulation with open-vocabulary instructions. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16, pp. 89–106 (2020)
25. Liu, X., Gong, C., Wu, L., Zhang, S., Su, H., and Liu, Q.: FuseDream: Training-Free Text-to-Image Generation with Improved CLIP+GAN Space Optimization, (2021). arXiv: 2112.01573v1 [cs.CV]
26. Matena, M., and Raffel, C.: Merging Models with Fisher-Weighted Averaging, (2021). arXiv: 2111.09832v1 [cs.LG]
27. Michel, O., Bar-On, R., Liu, R., Benaim, S., and Hanocka, R.: Text2Mesh: Text-Driven Neural Stylization for Meshes, (2021). arXiv: 2112.03221v1 [cs.CV]
28. Mitchell, E., Lin, C., Bosselut, A., Finn, C., and Manning, C.D.: Fast Model Editing at Scale, (2021). arXiv: 2110.11309v1 [cs.LG]
29. Mordvintsev, A., Olah, C., and Tyka, M.: DeepDream - a code example for visualizing Neural Networks, (2015). https://ai.googleblog.com/2015/07/deepdream-code-example-for-visualizing.html
30. Murdock, R.: "The Taming Transformers decoder really just goes! And this is with very little work." https://twitter.com/advadnoun/status/1367556678896394240
31. Murdock, R.: "Working on using the RN50x4 version of CLIP with the Taming Transformers VQGAN." https://twitter.com/advadnoun/status/1368081153375105027
32. Nam, S., Kim, Y., and Kim, S.J.: Text-Adaptive Generative Adversarial Networks: Manipulating Images with Natural Language. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.) Advances in Neural Information Processing Systems, pp. 42–51. Curran Associates, Inc. (2018)
33. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M.: GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models, (2021). arXiv: 2112.10741v3 [cs.CV]
34. Ntavelis, E., Romero, A., Kastanis, I., Van Gool, L., and Timofte, R.: SESAME: semantic editing of scenes by adding, manipulating or erasing objects. In: European Conference on Computer Vision, pp. 394–411 (2020)
35. Oord, A. van den, Vinyals, O., and Kavukcuoglu, K.: Neural Discrete Representation Learning. In: Advances in Neural Information Processing Systems, pp. 6309–6318. Curran Associates, Inc. (2017)
36. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., and Lischinski, D.: StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2085–2094 (2021)
37. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I.: Learning Transferable Visual Models From Natural Language Supervision. In: Meila, M., and Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, pp. 8748–8763. PMLR (2021)

38. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I.: Zero-Shot Text-to-Image Generation. In: Meila, M., and Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, pp. 8821–8831. PMLR (2021)
39. Riba, E., Mishkin, D., Ponsa, D., Rublee, E., and Bradski, G.R.: Kornia: an Open Source Differentiable Computer Vision Library for PyTorch. In: 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 3663–3672 (2020)
40. Sayers, D., Sousa-Silva, R., Höhn, S., Ahmedi, L., Allkivi-Metsoja, K., Anastasiou, D., Beňuš, Š., Bowker, L., Bytyçi, E., Catala, A., Çepani, A., Chacón-Beltrán, R., Dadi, S., Dalipi, F., Despotovic, V., Doczekalska, A., Drude, S., Fort, K., Fuchs, R., Galinski, C., Gobbo, F., Gungor, T., Guo, S., Höckner, K., Láncos, P., Libal, T., Jantunen, T., Jones, D., Klimova, B., Korkmaz, E., Maučec Mirjam, S., Melo, M., Meunier, F., Migge, B., Mititelu Verginica, B., Névéol, A., Rossi, A., Pareja-Lora, A., Sanchez-Stockhammer, C., Şahin, A., Soltan, A., Soria, C., Shaikh, S., Turchi, M., and Yildirim Yayilgan, S.: The Dawn of the Human-Machine Era: A forecast of new and emerging language technologies. (2021)
41. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 618–626 (2017)
42. Sharir, O., Peleg, B., and Shoham, Y.: The Cost of Training NLP Models: A Concise Overview. (2020)
43. Shocher, A., Gandelsman, Y., Mosseri, I., Yarom, M., Irani, M., Freeman, W.T., and Dekel, T.: Semantic Pyramid for Image Generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7457–7466 (2020). DOI: 10.1109/CVPR42600.2020.00748
44. Simonyan, K., Vedaldi, A., and Zisserman, A.: Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, (2014). arXiv: 1312.6034v2 [cs.CV]
45. Snell, C.: Alien Dreams: An Emerging Art Scene, (2020). https://ml.berkeley.edu/blog/posts/clip-art/
46. Tian, Y., and Ha, D.: Modern Evolution Strategies for Creativity: Fitting Concrete Images and Abstract Concepts, (2021). arXiv: 2109.08857v2 [cs.NE]
47. Tsimpoukelli, M., Menick, J., Cabi, S., Eslami, S.A., Vinyals, O., and Hill, F.: Multimodal Few-Shot Learning with Frozen Language Models. In: Advances in Neural Information Processing Systems (2021)
48. Underwood, T.: Mapping the latent spaces of culture, (2021). https://tedunderwood.com/2021/10/21/latent-spaces-of-culture/
49. Wang, Z., Liu, W., He, Q., Wu, X., and Yi, Z.: CLIP-GEN: Language-Free Training of a Text-to-Image Generator with CLIP, (2022). arXiv: 2203.00386v1 [cs.CV]
50. Wu, H.-H., Seetharaman, P., Kumar, K., and Bello, J.P.: Wav2CLIP: Learning Robust Audio Representations From CLIP, (2021). arXiv: 2110.11499v2 [cs.SD]
51. Yang, Z., and Buehler, M.J.: Words to Matter: *De novo* Architected Materials Design Using Transformer Neural Networks. Frontiers in Materials 8, 417 (2021)
52. Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., and Lipson, H.: Understanding Neural Networks Through Deep Visualization, (2015). arXiv: 1506.06579v1 [cs.CV]