A Author Contributions

Katherine Crowson: Originated the idea of combining VQGAN with CLIP and lead the development of the approach.

Stella Biderman: Lead the experimentation and writing of the paper.

Daniel Kornis: developed latent vector regularization and carried out experiments on Open-Edit.

Dashiell Stander: Assisted with generative experiments and the writing of the paper.

Eric Hallahan: Assisted with generative experiments and the writing of the paper.

Louis Castricato: Developed semantic image editing and masking techniques.

Edward Raff: Advised the project, proposed ablation experiments, and assisted in the writing of the paper.

B Additional Related Work

B.1 Image Generation and Editing

While there is a significant literature on semantic image generation and editing beyond those cited in the main body of the paper, it is overwhelmingly work that imposes strong constraints on and assumptions about the inputs it work with. Closed domain work such as Dong et al. [9], Nam, Kim, and Kim [32], Li et al. [23], and Patashnik et al. [36] achieves much higher quality images, but comes at the cost of being afunctional outside the domain the model was trained on. Other authors simplify the problem by using non-textual auxiliary inputs [43, 34] or filtering the inputs to a small, pre-defined vocabulary [17].

B.2 CNN Visualisation and Interpretability

Our approach is surprisingly similar to older work on visualizing and interpreting convolutional neural networks. Techniques for identifying what region(s) of an image lead to it being assigned a label by a classifier such as Simonyan, Vedaldi, and Zisserman [44], Selvaraju et al. [41], and Yosinski et al. [52] operate on similar principles to our backpropagating of the CLIP loss, and DeepDream [29] uses the same core idea of iteratively updating an image to match a label.

B.3 Building Multimodal Models Cheaply

Due to the high cost of training large, state-of-the-art transformer models [42, 4], methodologies that allow one to build off of previously trained models cheaply are highly valuable. The bulk of the existing literature on this topic focuses on editing or updating existing (typically unimodal) models in both computer vision [3] and natural language processing [7, 28, 26].

Another thread of research aims to leverage prompting to efficiently produce multimodal models from unimodal ones. Tsimpoukelli et al. [47] shows that an input-dependent version of prompt tuning [22] enables them to convert a 7 billion parameter language model into a multimodal model by only training 68 million of the parameters. Building on their success, Eichenberg et al. [10] replace the prompt tuning with adapters [16] and incorporate CLIP embeddings, enabling them to train a state-of-the-art visual question answering transformer at a fraction of the cost of doing so from scratch. Our work continues in this theme by removing the training entirely, at the cost of a modest increase in generation time.

C Observations of Public Use

As our models have been public for nearly a year at time-of-writing, we have had the opportunity to study how people use VQGAN-CLIP in the wild. While a systematic study is far outside the scope of this paper, we summarize our main observations in the hopes of guiding future human-computer interaction investigations. These observations are largely informed by casual conversations and interviews with users of our methodology, and by *in situ* observation of people interacting with the model. For further discussion of the HCI and sociological influences on generative modeling, we direct an interested reader to Snell [45], Underwood [48], and Ali and Parikh [1].

Human-AI Co-Creation The overwhelming feedback we have gotten from users is that they view VQGAN-CLIP not as an AI working in a vacuum but as an AI and a human working together to generate art. Users have also created new (and unintended) ways of working with VQGAN-CLIP that increase their direct agency over the art created. The most prominent example of this is that many users halt generation early to edit the partial generations directly, before restarting generation from their newly edited image. In this way users can remove artifacts, discourage undesirable components of an image, and try alternative paths through the optimization landscape.

Notebook-based development The overwhelming majority of people who have picked up and iterated on the methodologies described in this paper use notebooks such as Jupyter Notebook and Google Colab to do their work, instead of traditional repositories and scripts. Based on our conversations with users, the cause of this appears to be two-fold: widespread popularity among people who do not identify as computer scientists or developers and a lack of the necessary

computational resources to run the model locally. Similar considerations drive the popularity of websites that offer VQGAN-CLIP-as-a-service and the bot in our Discord server.

Iterated prompting While the typical methodology in machine learning for improving results from an algorithm focus on improvements to the training data or algorithm, our users have exhibited a strong preference for modifying the *prompts* they provide instead. We view this as exemplifying "natural language as an API," [40] wherein the actual mapping between the images and the text is less important than the fact that text is a natural medium for human interaction. The end-goal is to obtain desirable images *using whatever text input is necessary.*

D Additional Generations

The following is a selection of artwork generated using our technique by people other than the authors of this paper, often with highly complex prompting. We include them both to emphasize the diversity of artistic styles VQGAN-CLIP is capable of, as well as its widespread adoption. These images have served as the covers of academic journals, sold for thousands of dollars, formed the basis for physical paintings, and been displayed in art galleries.

E Additional Comparisons

The following images showcase the minDALL-E [19] and GLIDE (filtered) [33] generations with the prompts found in Figure 3.

(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

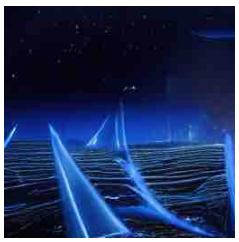
(g) an astronaut in the style of van Gogh

(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

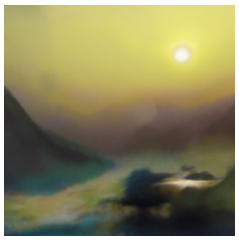


(l) a futuristic city in synthwave style

Figure 9: Example minDALL-E generations and their text prompts.

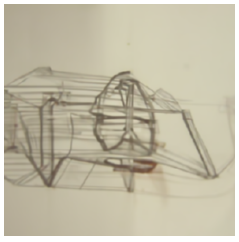
(a) Oil painting of a candy dish of glass candies, mints, and other assorted sweets

(b) A colored pencil drawing of a waterfall



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

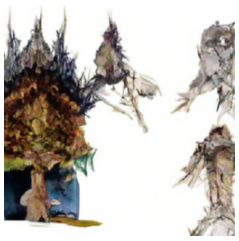
(d) A beautiful painting of a building in a serene landscape



(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 10: Example GLIDE (CLIP-Guided) generations and their text prompts.

(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

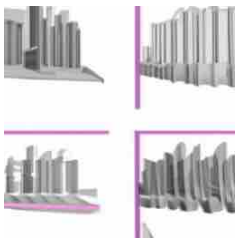
(g) an astronaut in the style of van Gogh

(h) Baba Yaga's house + fantasy art

(i) pickled eggs, tempera on wood

(j) effervescent hope

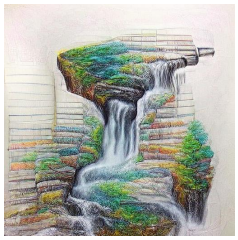
(k) the Tower of Babel by J.M.W. Turner



(l) a futuristic city in synthwave style

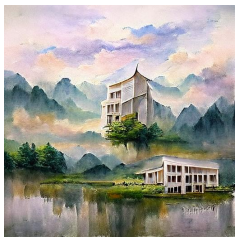
Figure 11: Example GLIDE (CF-Guided) generations and their text prompts.

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

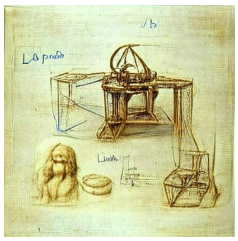


(b) A colored pencil drawing of a waterfall

(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



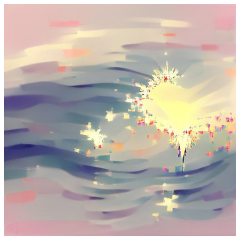
(e) sketch of a 3D printer by Leonardo da Vinci

(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 12: Example VQGAN-CLIP generations and their text prompts. This is the same figure as Fig. 2 repeated for ease of comparison with Figs. 9 to 11.

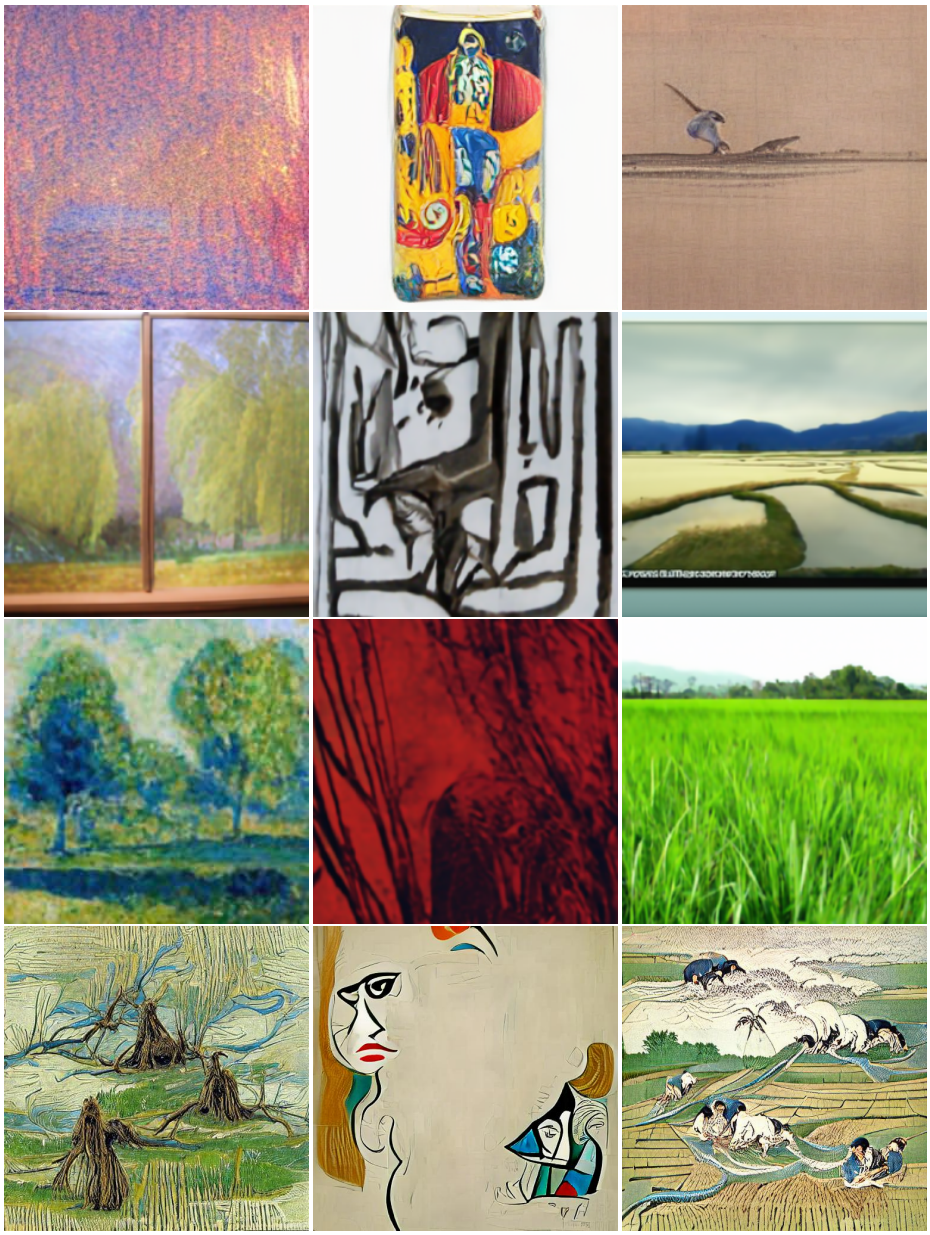
F Comparison of Artistic Impressions

We find that minDALL-E and both GLIDE (filtered) models have a much more tenuous understanding of famous artists than VQGAN-CLIP does. To showcase this we invent six fictional but plausible painting titles by each artist in Figure 4 and prompt the models to generate, e.g., "Willow Trees by van Gogh." While VQGAN-CLIP is able to consistently produce something that is recognizably in the style of the artist and containing the subject, we find that the other models frequently fail to produce artwork that matches the prompt, either in subject or in style.

We note that the GLIDE (filtered) models will frequently produce photorealistic images, despite the fact that none of the artists produce photorealistic artwork. As mentioned in Section 4, our primary goal is to produce *high quality art* rather than photorealistic images. Consequently, GLIDE's Hokusai generations - despite undoubtably being images of rice fields - rate lowly in our estimation as they have nothing in common with Hokusai's artwork. GLIDE's best results are in response to the van Gogh prompt in which the models clearly generate impressionistic images of trees, though the art resembles the work of Nicholas Verrall (in the case of CLIP-guided) and William Langson Lathrop or Claude Monet (CF-guided) far more so than van Gogh.

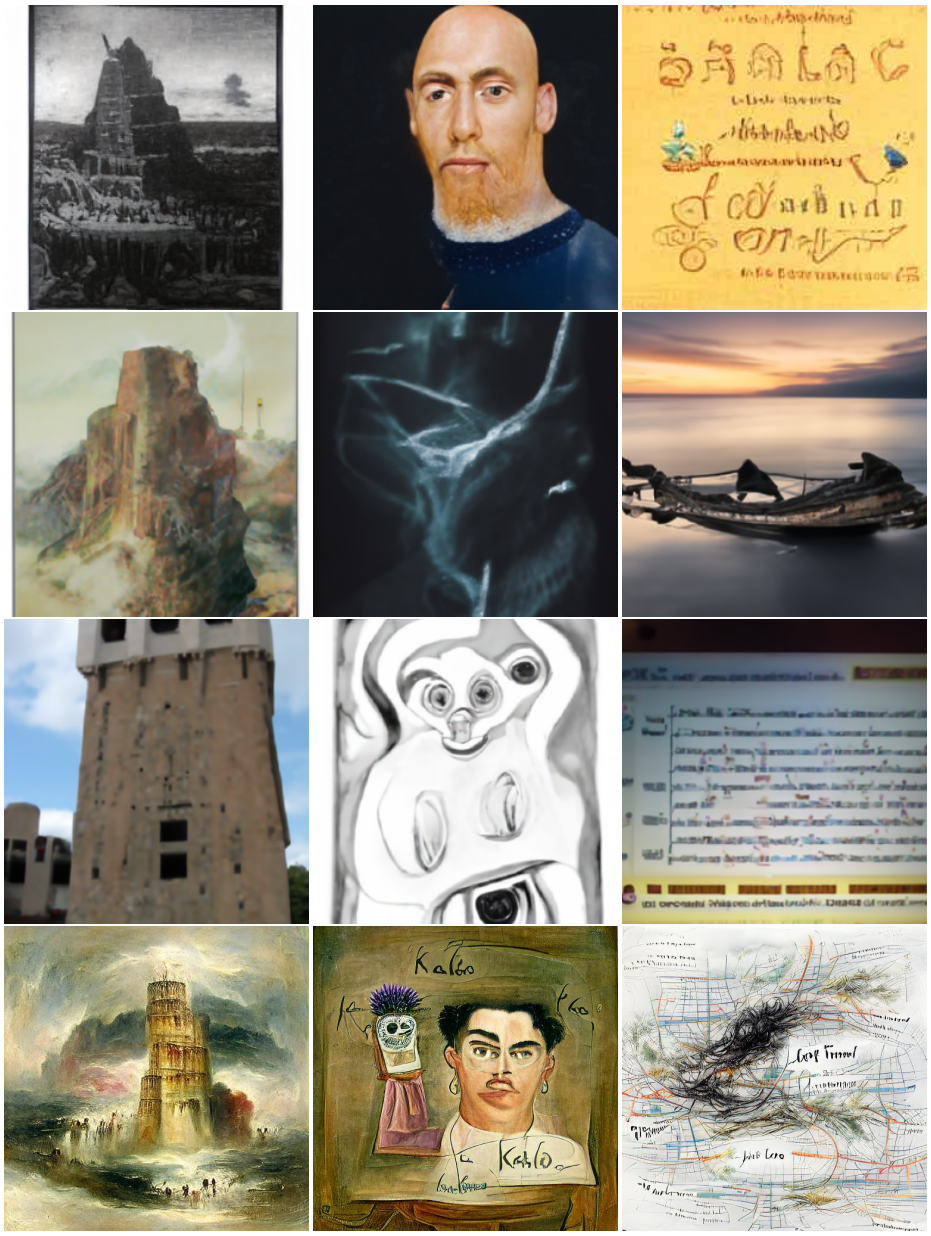
minDALL-E appears to be noticeably better at producing art, it frequently fails to agree with the prompt in both subject and in style. Perhaps most jarringly, in response to "A Self-Portrait by Kahlo" minDALL-E generates what is clearly a painting of a person, but the subject is a white man instead of a Latin woman.

As before, all images are cherrypicked best-of-five except for VQGAN-CLIP which is uncherrypicked. We omit ablations over various wordings of the prompt as our exploratory testing indicated that it would effect the overall quality of the generation but not the extent to which they match the prompts.



(a) Willow Trees by van Gogh (b) The Woman by Picasso (c) Rice farming by Hokusai

Figure 13: Attempts to generate novel works of art by famous artists. Top to bottom: minDALL-E, GLIDE (CLIP-guided), GLIDE (CF-guided), and our VQGAN-CLIP. While VQGAN-CLIP's generation quality varies, it is the only model that seems to have grasped the desired task.



(a) the Tower of Babel by Turner (b) A Self-Portrait by Kahlo (c) In Search of Lost Time by Mehretu

Figure 14: Attempts to generate novel works of art by famous artists. Top to bottom: minDALL-E, GLIDE (CLIP-guided), GLIDE (CF-guided), and our VQGAN-CLIP. While VQGAN-CLIP's generation quality varies, it is the only model that seems to have grasped the desired task.

G Ablations on Components

G.1 Latent Vector Regularization

Prior versions of this work used a methodology called Codebook Sampling, which optimizes a categorical distribution over a grid superimposed on the latent vectors. We found this approach was too slow for interactive use, and did leave considerable room for visual improvement. In Fig. 15 we show the improvement in quality of our current approach compared to the prior. By adding a regularization term Section 2.4, we obtain a methodology that is both faster and produces higher quality images than Codebook Sampling, giving the results presented in this paper. Our approach to regularization is incompatible with Codebook Sampling because of the tendency of Codebook Sampling to commit to particular codes early in training before the effects of regularization manifest.

(a) a forest rendered in the Unreal Engine

(b) a watercolor painting of the universal library

(c) An oil painting of The New York City Skyline by Natalia Goncharova

(d) the gateway between dreams trending on ArtStatio

Figure 15: Comparisons between codebook sampling (top), z-quantize (middle), and z-quantize with MSE regularization methods (bottom). Notice that z-quantize with regularization is able to produce finer details than alternative options, while non-regularized and codebook tend to yield muddled images.

G.2 Number of Augmentations

In Fig. 16 we ablate the number of augmentations to show their importance in consistent result quality.

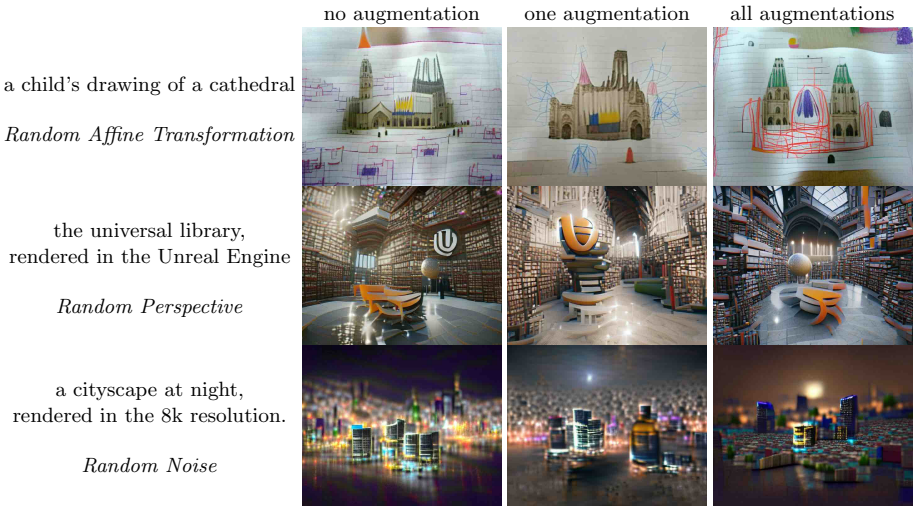


Figure 16: Examples of the impacts of various augmentations. Left the right: unaugmented, only the named augmentation, all augmentations.

We find that the affine augmentation reduces clutter and unwanted duplicative generations, prospective augmentation improves the consistency of the 3D geometry, the noise augmentation improves the isolation of the foreground from the background.

H Additional Components

H.1 Prompt Addition

One topic of immense interest for text-to-image models is *compositionality*: the extent to which the model is able to take multiple discrete concepts and combine them. While a detailed analysis of compositionality in VQGAN-CLIP is outside the scope of this paper, we have observed that VQGAN-CLIP is able to intelligently combine multiple prompts. By providing two separate text inputs and averaging the loss, we can effectively "add" the two prompts together. In Fig. 17 we show that adding a content and a style prompt results in an image that is quite similar to one produced by combining the style and content into a single prompt with natural language. We view this as a rich and exciting area for future research.

(a) A baseball game

(b) A watercolor painting of a baseball game

(c) A watercolor painting + A baseball game

(d) A watercolor painting

(e) A farm

(f) A charcoal drawing of a farm

(g) A charcoal drawing + A farm

(h) A charcoal drawing

Figure 17: A comparison of natural semantic combination and addition in the latent space. Left to right: the original prompt, a natural language prompt, latent space addition of prompts, and the descriptive modifier.

H.2 Masked Image Editing

Given a source phrase (text) s_p and a target phrase (text) t_p, we aim to construct an optimization rule that replaces all instances of s_p within a source image s_i with t_p, resulting in a target image (which is unknown at the beginning) t_i.

To generate t_i, we first need to mask the part component of our source image that corresponds to s_p, for instance if s_p is "Dog," we need to mask the part of the image containing a dog. We can utilize CLIP in a zero shot setting to perform masking as follows:

1. Crop s_i into a number of smaller sub images, $\tilde{S} = \{S_i^k\}_{k=1}^N$
2. $\forall i \in \tilde{S}$, compute $L(i) = f(i) \cdot g(s_p)$, where $f(\cdot)$ denotes our image encoder and $g(\cdot)$ denotes our text encoder.
3. Record the value of L(i) at the center of the crop of i
4. Normalize the resulting grey scale image, this is now our mask

We can then threshold this mask in order to determine which components of the mask actually contain our object of interest. In practice, we compute the threshold as two standard deviations below the average weight.

During generation, we crop and augment our current generated image and for every crop we compute the distance of the embedding of this crop to the embedding of the original image. This is used to preserve some notion of structure

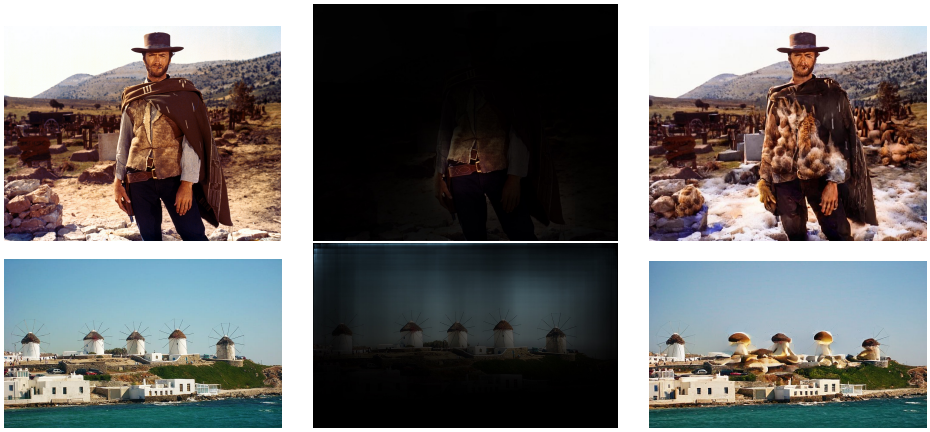


Figure 18: Two examples of masked editing, showing Jacket → Fur and Windmills → Mushrooms.

that was present in the original image. Similarly, we compute the distance of this embedded crop against the target phrase.

By minimizing a weighted sum of these distances, we can perform in-image object replacement and editing. See Figure 18.