clip2latent: Text driven sampling of a pre-trained StyleGAN using denoising diffusion and CLIP

Justin N. M. Pinkney
justin@lambdal.com
Lambda, Inc
San Francisco, USA

Chuan Li
c@lambdal.com

Abstract

We introduce a new method to efficiently create text-to-image models from a pre-trained CLIP and StyleGAN. It enables text driven sampling with an existing generative model without any external data or fine-tuning. This is achieved by training a diffusion model conditioned on CLIP embeddings to sample latent vectors of a pre-trained StyleGAN, which we call *clip2latent*. We leverage the alignment between CLIP's image and text embeddings to avoid the need for any text labelled data for training the conditional diffusion model. We demonstrate that clip2latent allows us to generate high-resolution (1024x1024 pixels) images based on text prompts with fast sampling, high image quality, and low training compute and data requirements. We also show that the use of the well studied StyleGAN architecture, without further fine-tuning, allows us to directly apply existing methods to control and modify the generated images adding a further layer of control to our text-to-image pipeline.

Figure 1: Images generated from text prompts by clip2latent trained on: Top: 1024x1024 StyleGAN2 FFHQ model. Bottom: 256x256 StyleGAN3 LHQ model. Prompts are given next to images and all were prefixed with "A photograph of".

1 Introduction

We take inspiration from the recent work of DALL-E 2[31] which trains a denoising diffusion model[12, 39, 40] to generate CLIP image embeddings from CLIP text embeddings. We generalise this approach to train a diffusion model to generate latent codes for a generative model conditional on CLIP image embeddings.

In effect, we map between the latent spaces of two different pre-trained models to enable controllable generation from a previous unconditional generative model. In this case we choose to generate StyleGAN2/3[15, 16] latent codes conditional on CLIP[29] image/text embeddings to enable text conditional synthesis using the pre-trained StyleGAN model. As both the text encoder and image synthesiser are frozen this method is quick to train, and by leveraging the shared embedding space of images and text in CLIP we are able to train it with no external data.

At inference time our model generates high-quality and high-resolution outputs, well aligned to a text prompt, with minimal images artefacts. We also show that we can benefit from the hierarchical and controllable nature of StyleGAN's existing latent space which can further be used to exert control over the generated images, see Section 5.2.

In summary, our main contributions are:

- A conditional diffusion model that connects pre-trained CLIP and StyleGAN models into a text-to-image model.
- A training scheme that requires no text-image pairs.
- A model which is quick to train, and can produce mega-pixel samples in under a second.

Source code and trained models used in this work are available at the following url: https://github.com/justinpinkney/clip2latent.

2 Related work

2.1 Text-to-image generation

Recent months have seen a tremendous advance in the quality of text-to-image generative models. The majority of these models are trained using large amounts of text-image paired data where, image generation is conditioned on the text input. To expedite the training and improve the quality of generation, several of the state of the art approaches make use of pre-trained models. Some use a pre-trained text encoder to leverage the semantic knowledge captured by large-scale text[34] or multi-modal pre-training[31]. Others have used a pre-trained image auto-encoder to reduce the effective spatial dimensionality required for acceptable resolution[10, 33] and in some case to work with an image representation more amenable to the image synthesis model[7, 30]. In contrast to our work however, the actual image synthesis model rarely makes use of an entirely pre-trained generator.

Early work in leveraging only pre-trained models was inspired by feature visualisation approaches[25], applying a variety of image parameterisations and regularisation allowed a successful iterative optimisation to maximise the similarity of an image to a given text prompt[20]. As well as conventional image parameterisations, this approach has also been applied to generative models where the latent code is directly optimised to produce model

outputs which best match a given text prompt[21]. This approach is conceptually closest to ours, where we replace the time consuming iterative optimisation with sampling from a conditional diffusion model which generates latent codes.

Another notable approach which leverages only pre-trained models is CLIP guided diffusion, where the classifier guidance approach for sampling a diffusion model can be applied using CLIP as the classifier[6]. However, this classifier guidance approach has since been shown to be limited by the lack of pre-trained CLIP models trained on noised images and both approaches are outperformed by classifier-free guidance for text-to-image generation[23].

2.2 Text driven manipulation of pre-trained StyleGAN

Although there has been limited work in using a pre-trained StyleGAN purely for text-to-image generation, there has been exploration of text driven *editing* using StyleGAN models. StyleCLIP[26], CLIP2StyleGAN[2], StyleMC[18], and others[8, 42, 46] use a variety of methods to extract editing directions in the latent space of a pre-trained StyleGAN based on text descriptions via CLIP. Although not the main application of our work, we note that our approach can also be applied to discovery of edit directions and show a small proof-of-concept example in Appendix F.

Finally, StyleGAN-Nada[9] performs domain adaptation of a pre-trained StyleGAN model by fine-tuning the generator based on a text description of the desired domain encoded by CLIP. Mind the Gap[49] takes a similar approach, but uses CLIP image encodings to perform domain adaptation to a new visual style. Unlike our approach this requires fine-tuning of the generator which modifies its entire output to match a single text/image prompt without adding any further text based control.

2.3 Language-Free training

One of our key contributions is the ability to train an text-to-image model without any text labelled training data. Two previous works have shown similar results. LAFITE[47] trains a GAN conditioned on CLIP image embeddings, we compare our results to their language-free text-to-image model in Section 5. Note the method of LAFITE requires training of the generator from scratch, whereas our method freezes the pre-trained StyleGAN generator. CLIP-GEN[45] similarly trains an auto-regressive transformer model to predict VQGAN tokens conditioned only on CLIP image embeddings and show that this approach generalises to CLIP text embeddings. In essence our method takes this, and similar[30, 33], approaches to the extreme: predicting a single StyleGAN latent vector rather than VQGAN tokens. This enables our model to be far smaller (48M vs 307M-1.6B parameters) and train much faster (500k vs 2M iterations) in exchange for being constrained closely to the original domain of the generative model.

3 Approach

Our approach is to map the latent spaces of pre-trained CLIP and StyleGAN models using a diffusion model we call clip2latent.

StyleGAN is a powerful generative model which can be sampled with a normally distributed latent vector $z \in \mathbb{R}^{512}$. From this vector StyleGAN applies a 'mapping network'

to map to a second latent space, $w \in \mathbb{R}^{512}$, finally this w vector is used to generate an image. To generate our dataset we randomly sample z latents and generate images from these with StyleGAN. Next we use the image encoder component of CLIP to encode StyleGAN generated images into the CLIP embedding space $e_i \in \mathbb{R}^{512}$. After encoding the StyleGAN generated images using CLIP, we can discard the generated images and original z latents to leave our training data (w, e_i) tuples, see Figure 2a.

We then train the clip2latent model using the same approach as the diffusion based prior from DALL-E 2[51], Figure 2b. Briefly we train a Gaussian diffusion model to generate w vectors conditioned on e_i embeddings. As in DALL-E 2 we train a transformer based model on a sequence consisting of: the CLIP image embedding e_i, an embedding representing the timestep in the diffusion process, the noised w latent vector, and a final learned embedding whos output predicts the denoised w.

Once we have trained clip2latent, we rely on the fact that CLIP can embed images and text into a shared latent space. At inference time we generate a CLIP embedding for a text description (e_t) and use this as the conditioning to generate a StyleGAN latent vector, from which we can create a high-resolution image using the StyleGAN generator, illustrated in Figure 2c.

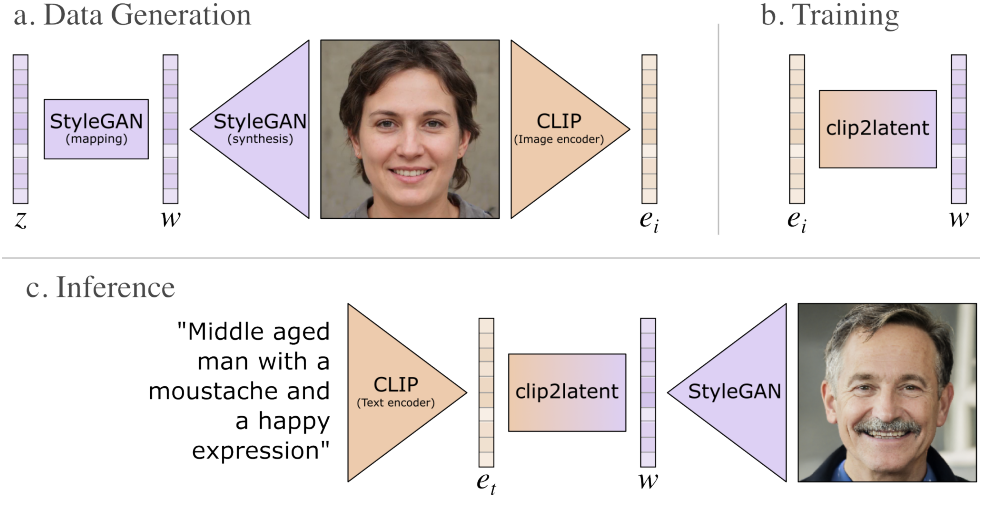

Figure 2: A schematic diagram of: a. synthetic data generation b. clip2latent diffusion model training, and c. text-to-image inference. Triangle and rectangle blocks indicate models, vertical striped bars indicate latent vectors or embeddings. z and w are the StyleGAN z and w latent spaces, e_i and e_t denote the CLIP embeddings space for images and text respectively.

4 Training details

4.1 Data generation

To generate paired training samples without the need for external images and text labels we use the pre-trained StyleGAN 2 model (config-F) trained on FFHQ[15]. We sample 1 million random latent vectors with no truncation and generate the CLIP embeddings for each

image using the ViT-B/32 pre-trained CLIP[29]. We resize the generated StyleGAN image to 224x224 pixels and apply the CLIP image normalisation. For each image we generate a single embedding and do not perform any data augmentation on the generated images.

4.2 Latent prior training

We then train a denoising diffusion model to generate the latent vectors conditioned on the CLIP image embeddings. We choose the same architecture as the prior in DALLE-2[31] and use the open source implementation DALL-E 2-Pytorch[43]. Our diffusion model is analogous to that presented in DALLE-2 where the CLIP image embeddings are replaced by the StyleGAN latent codes and the CLIP text embedding are replaced by the corresponding CLIP image embeddings, as such we could consider this model to be a *latent prior*.

During training we substitute the image embeddings with zeros with a probability of 0.2 to allow us to use classifier-free guidance[11] during inference. Rather than directly use the w latent vectors for training we subtract the mean and divide by the estimated standard deviation of samples from w space to better match the variance expected by the diffusion model. This also aligns with the common practice of learning StyleGAN latent prediction models relative to the mean latent vector[32].

To improve the ability of the model to generalise to text embeddings and avoid overfitting on image embeddings we take the approach introduced in LAFITE[47] where "pseudo-text" embeddings are generated from image embeddings using scaled Gaussian noise, see Section 5.1 and Appendix A.1.

We train our diffusion model for 1 million iterations at a batch size of 512, for full hyperparameter details see Appendix A. We select the best checkpoint as judged by generating 4 samples each for a set of 16 text prompts and computing the mean CLIP cosine similarity score between the text embedding and the CLIP image embedding of the generated image. In practice, we find that this peak validation score generally occurs at around 500 thousand iterations which takes approximately 9 hours on a single A100-80GB GPU.

4.3 Text-to-image synthesis

At inference time we generate the CLIP embedding for a text input and use this as the condition for the clip2latent network. By performing denoising diffusion, the clip2latent network generates a StyleGAN latent code which can then be used for image synthesis. During the denoising process we optionally employ timestep respacing[22] and super conditioning[11] to decrease the sampling time and increase the CLIP similarity respectively. We do not employ any clipping or renormalisation of the denoised outputs or any of the recent method to reduce super-conditioned artefacts[17, 44]. In practice, with our shortest denoising schedule, end-to-end text to image synthesis can be performed in under a quarter of a second on a single A100 GPU for a single image (see Table 1). As our image generation pipeline is fast and scales well to batching, we employ CLIP re-ranking to obtain the best CLIP similarity scoring sample from a batch of 16 candidates.

5 Results

Our method produces high quality, high resolution images based on text prompts without the need for paired image-text data during training. Figure 3 shows examples of text-to-image

Figure 3: Sample generations from our model (clip2latent) compared to direct optimisation of the StyleGAN latent vectors, LAFITE, and DALLE-2. For details of the comparison methods see Appendix C. Note, LAFITE generates samples at 256x256 pixels, whereas clip2latent and direct optimisation generate samples at 1024x1024. All prompts were prefixed with "A photograph of".

Table 1: CLIP similarity scores and inference times for our methods compared to previous work. Similarity scores are measured by generating samples for 64 text prompts and measuring the mean CLIP similarity score. For a full list of prompts see Appendix B, some samples could not be run in DALLE-2 due to the content policy. Inference times are measured using and A100-80GB PCIe GPU, for generation of a single sample including CLIP re-ranking and do not take advantage of any batching. We note the DALLE-2-pytorch library does not batch the two calls to forward when using classifier free guidance and could be optimised further. Inference times are not included for DALLE-2 as it is not available to run on comparable hardware.

Method	**CLIP score**	**run-time (s)**
clip2latent (ours)	0.316	11.760
clip2latent (ours) + timestep respacing	0.315	0.244
Direct Optimisation	0.321	19.191
LAFITE	0.278	0.045
DALLE-2	0.291*	—

generation via our method, compared against LAFITE and direct optimisation[24] of the w vector (for further details of the baseline comparison methods see Appendix C). We judge the similarity of generated image and text prompts using the standard CLIP similarity score, and show that our approach acheives CLIP socres close to optimisation methods whilst largely avoiding image artefacts and performing inference more rapidly, see Table 1.

5.1 Importance of embedding augmentation

Although CLIP encodes both images and text into a shared latent space it has been noted that the encodings for the two modalities are disjoint in this space[13, 19]. As we wish to train a text to image model with no external data we are restricted to using the CLIP image embeddings of StyleGAN generated images. Although, previous works has shown that simply training models conditional on image embeddings and substituting text embeddings at inference time is sufficient to generate good quality results [31, 45], we find the the addition of scaled Gaussian noise to the condition embedding vectors (as suggested in LAFITE[47]) is crucial in allowing the model to generalise across both image and text embeddings produced by CLIP.

Although the authors of LAFITE describe this as a method for producing "psuedo text encodings", we consider this to be more analogous to a method of data augmentation, noting that the "pseudo text" embeddings and original image embedding have a similarity score of around 0.8 (given a noise scaling of 0.75 using in LAFITE), far higher than any real text embedding with a well matched image embedding (typically in range 0.3-0.4[13]). Without the addition of noise, the model learns to produce latent codes corresponding very closely to the input image, but fails to generalise to text embeddings. At very high noise levels the conditioning is less informative and the model performance deteriorates. We find the optimal level of noise scaling to be 1.0, see Appendix Figure 1.

5.2 Making use of StyleGAN

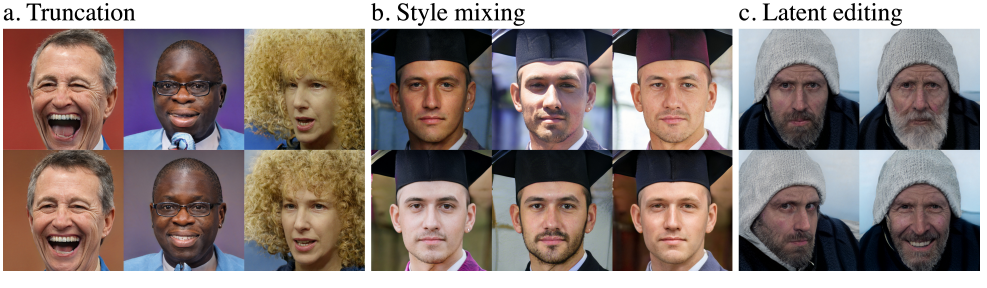

Figure 4: Demonstrating the application of existing StyleGAN editing techniques to add additional levels of control. a: Artefact correction using truncation, (top row original, bottom row truncation 0.8) for images of "a British politician laughing happily", "a Nigerian professor of economics", "person with very tight curly blonde hair". b: Style mixing to create variations of colour, lighting, and texture for "a university graduate". c: Age, Pose, and Smile editing using InterfaceGAN directions on a generated sample (top-left) from the prompt "an arctic explorer".

The image synthesis portion of our text-to-image pipeline uses the well studied and understood StyleGAN generator. This means we can leverage many of the well known and favourable properties of StyleGAN to exert further control over our generated images. In the following section we demonstrate how our method can take advantage of: truncation for artefact removal, style mixing for improving diversity, and the well-known editability of StyleGAN's latent space.

5.2.1 Truncation

We find that at high guidance scales our latent vectors can sometimes fall outside the typical domain of StyleGAN latent space, generating unnatural artefacts. One method to alleviate this is to use the well-known technique of truncation[4] to move the generated latent closer to the mean latent vector. This trades off image-text similarity and diversity for image fidelity, see Figure 4a. Although in some cases this can actually *increase* the CLIP similarity score as reducing artefacts brings the image closer to the domain of real face image.

5.2.2 Style Mixing

We can also take advantage of the extended latent space of StyleGAN ($w+$) where the latent vector for each layer controls the generated image at a different resolution scale[15]. One straightforward way to utilise this structure is to add extra colour and lighting diversity to our generated images by performing Style Mixing with our generated latent vector for the lower resolution layers which control most of the semantic appearance of the generated image, with a randomly sampled latent for higher resolution layers which predominantly control colour and lighting, see Figure 4b.

5.2.3 Latent Editing

Finally we can also leverage the extensive literature on finding semantically meaningful edits in StyleGAN latent space to edit our generated images, adding an extra level of control to our text-to-image pipeline. This allows us to make use of the wealth of existing high-quality facial editing directions available for StyleGAN. We show the application of some well known InterfaceGAN[37] directions to our generated images in Figure 4c. It has previously been shown that high-quality edits require latent vectors to lie well within the typical domain of StyleGAN's latent space[41]. The generative nature of the clip2latent diffusion model ensures that text-generated latents are within the domain of StyleGAN without the need for additional losses such as the "latent discriminator" in *Tov et al.*[37]. This ensure that our method can generate high-quality edits with few artefacts, in contrast to alternatives such as direct optimisation which generates latents with poor editability, see Appendix Figure 2.

6 Discussion

We note our method is not restricted to StyleGAN in particular and in theory could be applied to any generative model. In practice our method appears to favour models with a relatively low dimensional latent space (see Appendix E) and clearly is well suited to those models for which high quality pre-trained networks are available. To demonstrate the generality of our method beyond faces we also train a text-to-image model using a 256x256 StyleGAN3 landscape model[27] trained on the LHQ dataset[38], see Figure 1, bottom row. Throughout

this work we have chosen the StyleGAN family of architectures due to the wide range of pre-trained models available[28] and its high-resolution, fast inference and state of the art performance in many domains[14, 35]. However, there is no reason a different GAN (e.g. BigGAN[4]) or entirely different class of generative model (e.g. VAE[5]) couldn't also be used.

Although we have chosen to take a simple data-free approach to training our model, which does not rely on any exist text-image paired data, there is now wide availability of such datasets across a variety of domains thanks to the open-source efforts of LAION[36]. To make use of such paired data would require embedding the real-world images into the latent space of StyleGAN for which there exist many methods[1, 3, 32]. We leave this as an avenue for exploration for future work.

Our generated images are constrained to lie withing the w space of StyleGAN, although this can produce a large variety of images its expensiveness is limited. Previous work has shown that the $w+$ space is vastly more capable in what images it can represent[1]. We show some limited exploration in applying our model to other StyleGAN latent spaces in Appendix E, but leave extensions to the other StyleGAN latent spaces to future work.

As well as generalisation to other generative model latent spaces we believe that the application of powerful diffusion models can allow the addition of conditional sampling to previously unconditional models based on any image encoding, for example facial recognition/attribute networks or other classification models. We look forward to future applications of diffusion models as tools for arbitrarily mapping between latent spaces of pre-trained models.

7 Conclusion

Our method allows sampling from a pre-trained generative model conditioned on CLIP image and text embeddings, providing a lightweight way to train and inference an effective text-to-image generation model without access to external data. Our method generates high quality results with satisfactory CLIP scores whilst largely avoiding image artefacts from earlier methods. We hope our work sheds a light on how the connection between text and image enables more powerful use of pre-trained generative models.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN: How to embed images into the StyleGAN latent space? April 2019.

[2] Rameen Abdal, Peihao Zhu, John Femiani, Niloy J Mitra, and Peter Wonka. CLIP2StyleGAN: Unsupervised extraction of StyleGAN edit directions. December 2021.

[3] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. ReStyle: A Residual-Based StyleGAN encoder via iterative refinement. April 2021.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. September 2018.

[5] Rewon Child. Very deep VAEs generalize autoregressive models and can outperform them on images. November 2020.

[6] Katherine Crowson. Clip guided diffusion hq 256x256., 2021. URL https://colab.research.google.com/drive/12a_Wrfi2_gwwAuN3VvMTwVMz9TfqctNj.

[7] Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. CogView2: Faster and better Text-to-Image generation via hierarchical transformers. April 2022.

[8] Aviv Gabbay, Niv Cohen, and Yedid Hoshen. An image is worth more than a thousand words: Towards disentanglement in the wild. June 2021.

[9] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. StyleGAN-NADA: CLIP-Guided domain adaptation of image generators. August 2021.

[10] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for Text-to-Image synthesis. November 2021.

[11] Jonathan Ho and Tim Salimans. Classifier-Free diffusion guidance. September 2021.

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv [cs.LG]*, June 2020.

[13] Travis Hoppe. Tweet, 2021. URL https://twitter.com/metasemantic/status/1356406256802607112.

[14] Minguk Kang, Joonghyuk Shin, and Jaesik Park. StudioGAN: A taxonomy and benchmark of GANs for image synthesis. June 2022.

[15] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. December 2019.

[16] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. June 2021.

[17] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *arXiv [cs.CV]*, June 2022.

[18] Umut Kocasari, Alara Dirik, Mert Tiftikci, and Pinar Yanardag. StyleMC: Multi-Channel based fast Text-Guided image generation and manipulation. December 2021.

[19] Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. March 2022.

[20] Ryan Murdock. Tweet, 2021. URL https://twitter.com/advadnoun/status/1348375026697834496.

[21] Ryan Murdock. Tweet, 2021. URL https://twitter.com/advadnoun/status/1351038053033406468.

[22] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. February 2021.

[23] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with Text-Guided diffusion models. December 2021.

[24] nshepperd. stylegan3 with clip guidance, 2021. URL https://colab.research.google.com/drive/1eYlenR1GHPZXt-YuvXabzO9wfh9CWY36.

[25] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11), November 2017.

[26] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: Text-Driven manipulation of StyleGAN imagery. March 2021.

[27] Justin N. M. Pinkney. Awesome Pretrained StyleGAN3, .

[28] Justin N. M. Pinkney. Awesome pretrained StyleGAN2, .

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. February 2021.

[30] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image generation. February 2021.

[31] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional image generation with CLIP latents. April 2022.

[32] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a StyleGAN encoder for Image-to-Image translation. August 2020.

[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution image synthesis with latent diffusion models. December 2021.

[34] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv [cs.CV]*, May 2022.

[35] Axel Sauer, Katja Schwarz, and Andreas Geiger. StyleGAN-XL: Scaling StyleGAN to large diverse datasets. February 2022.

[36] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. LAION-400M: Open dataset of CLIP-Filtered 400 million Image-Text pairs. November 2021.

[37] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. InterFaceGAN: Interpreting the disentangled face representation learned by GANs. May 2020.

[38] Ivan Skorokhodov, Grigorii Sotnikov, and Mohamed Elhoseiny. Aligning latent and image spaces to connect the unconnectable. April 2021.

[39] Jascha Sohl-Dickstein, Eric A Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. March 2015.

[40] Yang Song and Stefano Ermon. Improved techniques for training Score-Based generative models. June 2020.

[41] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for StyleGAN image manipulation. February 2021.

[42] Christos Tzelepis, James Oldfield, Georgios Tzimiropoulos, and Ioannis Patras. ContraCLIP: Interpretable GAN generation driven by pairs of contrasting sentences. June 2022.

[43] Phil Wang. Dall-e 2 - pytorch, 2022. URL https://github.com/lucidrains/DALLE2-pytorch.

[44] Tengfei Wang, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen. Pretraining is all you need for Image-to-Image translation. May 2022.

[45] Zihao Wang, Wei Liu, Qian He, Xinglong Wu, and Zili Yi. CLIP-GEN: Language-Free training of a Text-to-Image generator with CLIP. March 2022.

[46] Zipeng Xu, Tianwei Lin, Hao Tang, Fu Li, Dongliang He, Nicu Sebe, Radu Timofte, Luc Van Gool, and Errui Ding. Predict, prevent, and evaluate: Disentangled Text-Driven image manipulation empowered by Pre-Trained Vision-Language model. November 2021.

[47] Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. LAFITE: Towards Language-Free training for Text-to-Image generation. November 2021.

[48] Peihao Zhu, Rameen Abdal, Yipeng Qin, John Femiani, and Peter Wonka. Improved StyleGAN embedding: Where are the good latents? December 2020.

[49] Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Mind the gap: Domain gap control for single shot domain adaptation for generative adversarial networks. October 2021.

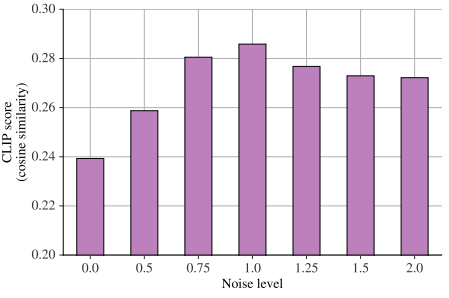

Figure 5: CLIP similarity score measured at different noise scaling factors (α) applied to the CLIP image embeddings on which the model is conditioned.

Appendices

A Hyperparameters

A.1 Noise level

Following LAFITE[47] we apply scaled Gaussian noise to generate an augmented CLIP embeddings, e'_i, as follows:

$$e'_i = \frac{y}{||y||_2}, \quad y = e_i + \alpha \frac{\varepsilon}{||\varepsilon||_2} \tag{1}$$

where e_i represents the normalised CLIP image embedding, $\varepsilon \sim \mathcal{N}(0, I)$ is a Gaussian noise sample of the same dimension as e_i, and α is the noise scaling factor.

To ascertain the optimal noise level for CLIP image embedding augmentations we perform a simple search of possible noise values. For each noise level we run training of clip2latent model using the StyleGAN2 FFHQ model for 100 million iterations. We then select the best model by generating images using a set of 16 text prompts and measuring the CLIP similarity score. We find that the optimal value for α is 1.0, see Figure 5

A.2 Model hyper parameters

Model and training hyper-parameters used for training both the StyleGAN2 FFHQ and StyleGAN3 LHQ clip2latent models. Parameter names follow the keys in the training configurations available in the code repository at: https://github.com/justinpinkney/clip2latent. In total our model has 48.9 million parameters.

B CLIP score prompts

Below is the list of prompts used for CLIP similarity scoring with clip2latent and comparable methods, all captiosn were prefixed by "A photograph of". An asterisk indicates the prompt could not be used with DALLE-2 due to the content policy applied by OpenAI.

Model parameters

Parameter	Value
timesteps	1000
beta_schedule	cosine
predict_x_start	True
cond_drop_prob	0.2
dim	512
depth	12
dim_head	64
heads	12

Training parameters

Parameter	Value
iterations	1,000,000
batch_size	512
lr	1.0e-4
weight_decay	1.0e-2
ema_beta	0.9999
ema_update_every	10
noise_scale (α)	1.0
Adam β_1, β_2	0.9, 0.999

1. a person with glasses
2. a person with brown hair
3. a person with curly blonde hair
4. a person with a hat
5. a person with bushy eyebrows and a small mouth
6. a person smiling
7. a person who is angry
8. a person looking up at the sky
9. a person with their eyes closed
10. a person talking
11. a man with a beard
12. a happy man with a moustache
13. a young man
14. an old man
15. a middle aged man
16. a youthful man with a bored expression
17. a woman with a hat
18. a happy woman with glasses
19. a young woman
20. an old woman
21. a middle aged woman
22. a baby crying in a red bouncer
23. a child with blue eyes and straight brown hair in the sunshine
24. an old woman with large sunglasses and ear rings
25. a young man with a bald head who is wearing necklace in the city at night
26. a youthful woman with a bored expression
27. President Xi Jinping *
28. Prime Minister Boris Johnson *
29. President Joe Biden *
30. President Barack Obama *
31. Chancellor Angela Merkel *
32. President Emmanuel Macron *
33. Prime Minister Shinzo Abe *
34. Robert De Niro
35. Danny Devito
36. Denzel Washington
37. Meryl Streep *
38. Cate Blanchett *
39. Morgan Freeman *
40. Whoopi Goldberg *
41. Usain Bolt *
42. Muhammad Ali *
43. Serena Williams *
44. Roger Federer *
45. Martina Navratilova *
46. Jessica Ennis-Hill *
47. Cathy Freeman *
48. Christiano Ronaldo *
49. Elsa from Frozen
50. Eric Cartman from South Park
51. Chihiro from Spirited Away
52. Bart from the Simpsons
53. Woody from Toy Story
54. a university graduate
55. a firefighter
56. a police officer
57. a butcher
58. a scientist
59. a gardener
60. a hairdresser
61. a man visiting the beach
62. a woman giving a TED talk
63. a child playing with friends
64. a person watching birds in the forest

C Comparison methods

C.1 Direct latent optimisation

We employ code by nshepperd[24] for optimisation of StyleGAN latent vectors based on CLIP similarity score to a text prompt. We modify the existing code such that optimisation is performed in w space (rather than the original $w+$ space) of StyleGAN to more closely match our approach. We otherwise leave optimisation parameters as originally shared in the notebook.

C.2 LAFITE text-free model

We use the text-free MM-CelebA trained model provided by the original authors of LAFITE. As the inference of LAFITE is fast, to most closely compare LAFITE with our method we employ the same CLIP re-ranking approach as clip2latent, and sample 16 generations for each prompt and select that with the best CLIP similiary score.

C.3 DALLE-2

We compare to DALLE-2 using the web interface provided by OpenAI. The web interface was returns 4 images per prompt, to avoid bias the first sample was taken for every prompt.

To try and generate images more comparable with the style of other methods trial and error was used to generate an engineered prompt of the form: "A photo headshot, single whole head centred, of ⟨*caption*⟩, flickr dslr, portrait photograph"

D Latent Editability

Our methods generate latent vectors which can effectively be edit using existing editing directions in StyleGAN's latent space. In contrast edits of the same magnitude applied to latents produced by direct optimisation frequently produce artefacts such as changes in composition, poor image quality, and in some cases complete failure to generate a recognisable face, see Figure 6.

Figure 6: Comparison of editability of latents generated from the text prompt "A photograph of an arctic explorer" generated using clip2latent vs direct optimisation. Edit directions were applied with the same magnitude to both generated latents.

E Other StyleGAN latent spaces

We also explored the training of clip2latent model using the extended latent spaces of StyleGAN. The $w+$ space has been shown to be more expressive then w and is frequently used to accurately embed arbitrary images into the StyleGAN latent space[1]. However generating samples from $w+$ by randomly combining w samples in from all 18 vectors gives rise to low quality generated images[18] making it unsuitable for our application.

We briefly explore the use of a more limited latent space comprising three independent w vectors which correspond to low, medium and high resolution layers. We term this latent space $w3$ and find that randomly generated samples are still high-quality but more diverse. We explored training a clip2latent model in this space using the same hyper-parameters as in

w space. We found that although the model trains successfully, the overall CLIP similarity scores are not improved and the model converges more slowly. We leave further exploration of how to exploit the greater diversity of extended latent spaces to future work.

F Finding edit directions

ALthough not our main application we also note that our latent generation method can be used to find directions within StyleGAN's latent space. By generating sets of latent vectors corresponding to a "positive" and "negative" prompt we can measure the direction between the two sets in order to obtain a latent direction, see Figure 7.

Figure 7: Examples of editing direction found using text based latent vector generation using clip2latent. Left column: random samples from StyleGAN. Following columns: edits in the directions: smile, curly hair, age, glasses, make up, beard.