CLIP-guided Zero-Shot Text-to-Image Generation

Linfeng Du
Department of Computer Science
University of Toronto
linfeng.du@mail.utoronto.ca

Kejia Yin
Department of Computer Science
University of Toronto
kejia.yin@mail.utoronto.ca

Xuduo Gu
Department of Computer Science
University of Toronto
xuduo.gu@mail.utoronto.ca

Abstract

CLIP model could encode images and texts into embeddings in a joint feature space, which makes it possible to practice guided image generation with only text prompts. Conditional GAN models are commonly used to generate images from latent noise and condition vectors. In this work, we use GAN to generate a set of images from randomly sampled noise and selected condition vectors. The text prompt and generated images are encoded using a pre-trained CLIP model, and the image whose CLIP embedding has the highest cosine similarity with the text embedding will be selected as the output. Our experiment finds that the best image quality is logarithmic to the total number of images generated. Our approach significantly boosts the efficiency of generating high-quality images than random GAN generation. Our code is available at `https://github.com/CapFreddy/CSC2516-Course-Project`.

1 Introduction

Contrastive Language-Image Pretraining (CLIP) closes the gap between text and image data by learning a joint embedding space for both modalities [1]. As a result, recent text-to-image generation methods exploit the CLIP space for conditional image generation and have achieved ground-breaking results, demonstrating good controllability over generated high-quality images [2]. However, the generative modelling approach adopted by the state-of-the-art methods renders them rather costly to train and requires a significant amount of paired annotation to yield satisfactory performance. In this paper, we propose an alternative framework for zero-shot text-to-image generation that actively exploits the cross-modality nature of CLIP embeddings and uses pre-trained conditional GANs as the image generator to avoid training at all. We leverage the text encoder to select the most relevant GAN labels to restrict the search space and leverage the image and text encoders to score the generated images in order the retrieve the most relevant ones. We study two types of pre-trained conditional GANs, Self-Attention GAN (SAGAN) and BIGGAN, and evaluate their generation performance under our framework in terms of CLIP score and FID as well as their generation efficiency through thorough experiments.

2 Related Work

After DALL·E 2 [2] rose to prominence in 2022, it gained an enormous amount of attention from both inside and outside the research community with its outstanding image generation capability. One key component of the DALL·E 2 model is CLIP [1], which provides robust feature representation for

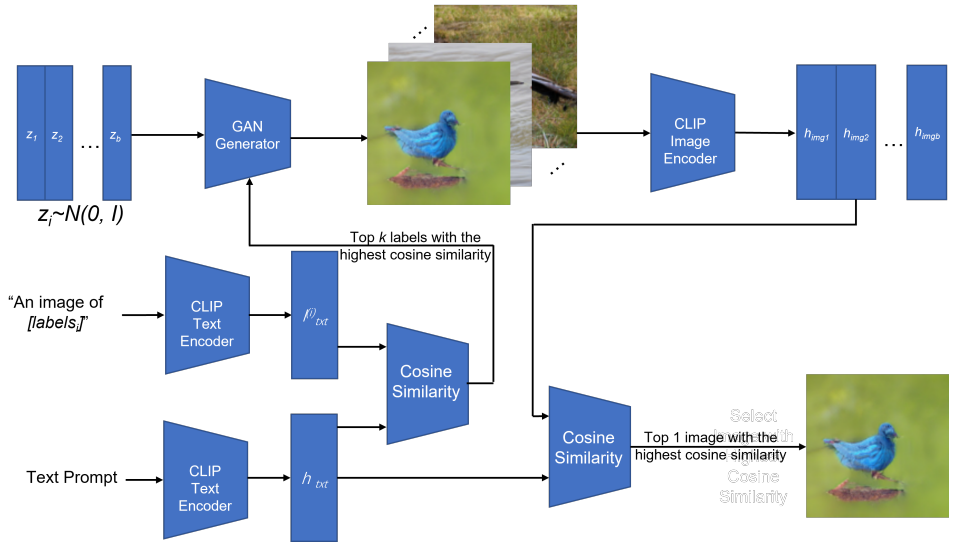

Figure 1: Overview of our proposed method.

both texts and images in the same space. Similar to DALL·E 2, our approach will also adopt CLIP as the bridge to connect images with texts. However, while DALL·E 2 utilizes diffusion models to decode the embeddings, our approach instead uses varied GAN models to generate images from the latent code. This is somehow similar to LAFITE [3], which uses StyleGAN2 and CLIP to realize language-free training to generate images from texts in a self-supervised manner. While LAFITE uses CLIP to decouple the training process with textual labels, our approach adopts CLIP in an opposite way to achieve image-independent learning for text-to-image generation.

There are also researches leveraging pre-trained GANs and CLIP-guided loss to perform text-to-image generation. CLIP-GLaSS [4] used a genetic algorithm to search the latent code in pre-trained GAN's latent space which can minimize the similarity loss between the CLIP embeddings of generated image and text prompt while also letting the generated image being able to fool the discriminator from real images. CLIP-GLaSS is similar to our approach but we are different in many aspects: 1) our approach does not need real images during training, 2) rather than optimizing one latent code for each text prompt, we simplify the searching process as per randoms search, based on the most relevant labels selected via textual similarity.

VQGAN and CLIP are also paired to perform image generation [5], where VQGAN serves as an autoencoder to generate the latent code on which gradient descent is performed. Meanwhile, VQGAN decodes the latent code to generate an image, which will later be randomly augmented and compared to the text prompt in terms of their embeddings in the latent space spanned by CLIP. Our approach is different from the VQGAN-CLIP model. Since our approach is image-free, no image encoder is needed to generate the latent code in the first place.

3 Method

As shown in Figure. 1, the first step is to use a pre-trained CLIP model to encode the text prompt into a vector h_{txt} in the latent feature space. Using the same CLIP text encoder, we encode the text "*An image of* $[labels_i]$" into $l_{txt}^{(i)}$, where $labels_i$ is the ImageNet [6] class label of the one-hot vector whose i-th entry is one. We then sort all $l_{txt}^{(i)}$ vectors in descending order according to the cosine similarity between $l_{txt}^{(i)}$ and h_{txt}. The one-hot vectors corresponding to the top k labels with the highest cosine similarity scores will be collected into a set $\mathcal{Y}$. Following that step, we randomly sample vectors z from a multivariate standard normal distribution and vectors y uniformly chosen from $\mathcal{Y}$, which will be later passed into a pre-trained GAN generator to generate the corresponding number of images. These images will be encoded by the CLIP model into vectors h_{img}. The cosine similarity between h_{txt} and h_{img} will be calculated as the CLIP score and the highest CLIP score

Algorithm 1 Proposed Method

Input text prompt $text$ and hyperparameters: k, b_{max}
Initialize pre-trained CLIP text encoder E_t and image encoder E_i, pre-trained GAN generator G
Prepare top k ImageNet labels as $\mathcal{Y}$ based on CLIP score
$MAXCLIP = 0, img_{best} = None$
for $b = 1, 2, 3, ..., b_{max}$ **do**
    Sample $z \sim \mathcal{N}(0, I)$ and $y \sim Uniform(\mathcal{Y})$
    Generate $img = G(z, y)$
    Compute $CLIPScore = CosineSim(E_t(text), E_i(img))$
    if $CLIPScore > MAXCLIP$:
        $MAXCLIP = CLIPScore, img_{best} = img$
return img_{best}

will be recorded for analysis. The FID score of these images will also be calculated as a comparison to the CLIP score.

We control the size of the total number of images generated b_{max} and the k parameter as the hyperparameters for this work. In the work, we experiment with all values b_{max} between 1 and 100. The upper bound of 100 for the b_{max} parameter is determined empirically since all models demonstrate convergence on the best CLIP score after b_{max} reaches 100. We pick the values of k in a pseudo-logarithmic manner to efficiently study the influence of heuristic refining of the class labels at different magnitudes.

4 Experiment

4.1 Dataset and Pre-trained Models

We use the Caltech-UCSD Birds-200-2011 [7] dataset to evaluate our proposed method. This dataset contains 11,788 images of 200 subcategories belonging to bird, 9010 for training and 2778 for testing following [8]. Since our method does not involve any training, we only use the test set in the following sections, except that we compute an average CLIP score on the training set.

To generate images, we used the pre-trained model released at `https://github.com/lukemelas/pytorch-pre-trained-gans` [9]. More specifically, we use SAGAN[10] and BIGGAN[11].

For the pre-trained CLIP model, we use the code released by OpenAI at `https://github.com/openai/CLIP` and the ViT-B/32 model for all the experiments below.

4.2 Best CLIP Score and FID

In Table. 1, we show the statistics of the best CLIP score and FID after generating 100 images with three different random seeds. Because the standard deviations of the best CLIP score are too small, they are not shown in this table, but they can be found in the Appendix A. Best CLIP score represents the similarity between generated image and caption text. We can see that SAGAN obtains the highest best CLIP score when $k = 10$, while $k = 20$ for BIGGAN. In addition, the best CLIP score increases first and then decreases as the k become larger, which is expected as when k is small, we may not include the most suitable class in the Top-K label set, and when k is large, it gets closer to purely random generation and is much less efficient. FID indicates how far away is the generated image distribution from the ground truth test image distribution. For SAGAN, the FID decreases first and then increases as k becomes larger, this trend matches the best CLIP score. However, for BIGGAN, the FID only decreases. By inspecting the generated images, we find that the variety of the images generated by BIGGAN is far less than those generated by SAGAN when k is small. Even though BIGGAN generates some images that are close to the caption and yield high CLIP scores, the whole image distribution may not be close to the test set. As k become larger, the variety of images increases and results in lower FID. Some generation examples can be found in Appendix B.

Method	k=1	k=10	k=20	k=50	k=100	k=1000
SAGAN	0.2986	**0.3113**	0.3104	0.3067	0.3028	0.2882
BIGGAN	0.2797	0.3045	**0.3053**	0.3027	0.2986	0.2819

Method	k=1	k=10	k=20	k=50	k=100	k=1000
SAGAN	$106.37 \pm .82$	$60.52 \pm .56$	**55.62** $\pm .97$	$56.14 \pm .65$	$57.11 \pm .65$	76.04 ± 1.43
BIGGAN	$117.51 \pm .16$	$95.89 \pm .20$	$93.52 \pm .58$	$90.66 \pm .47$	87.38 ± 1.07	**84.20** $\pm .85$

Table 1: Best CLIP score (upper) and FID (below)

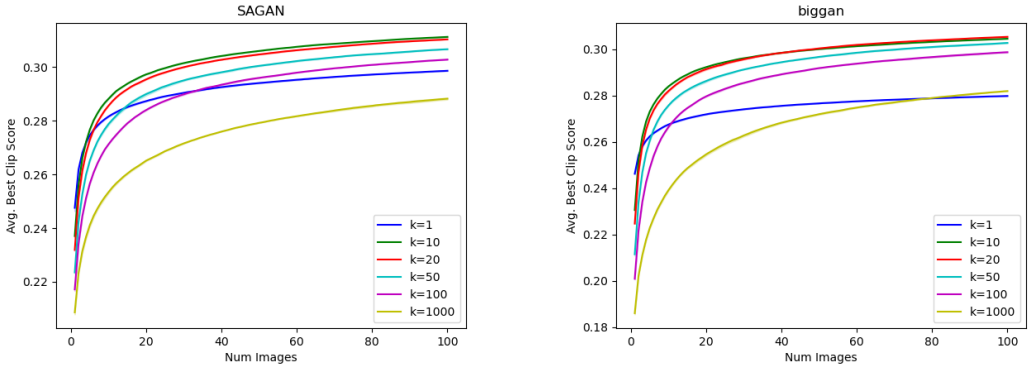

Figure 2: Best CLIP Score v.s. Number of generated images with different Top-K labels after generating 100 images

Method	k=1	k=10	k=20	k=50	k=100	k=1000
SAGAN	$13.67 \pm .58$	8	10	14.33 ± 0.58	20.67 ± 0.58	72.33 ± 2.52
BIGGAN	> 100	11	12	17.67 ± 0.58	27.67 ± 0.58	> 100

Table 2: Number of Images needed to be better than a train set in terms of CLIP score

4.3 Generation Efficiency

Since our method intrinsically relies on random generation, it is important to know how many images we need to generate before we find a good one. As shown in Figure. 2, we plot the average best CLIP score v.s. the number of generated images. We can see for both SAGAN and BIGGAN, the CLIP score increases most quickly when $k = 10$. However, we do not have a clear idea that how high the CLIP score should be when we consider an image to be a good one given the caption. Therefore, we compute the average CLIP score between images and captions on the train set and report how many images we need to generate to make the average best CLIP score greater than it in Table. 2 based on three random seeds. We can see that when $k = 10$, SAGAN achieves this threshold most efficiently which is 9 times more efficient than purely random generation. When $k = 1000$, BIGGAN fails to reach this threshold, but it is at least 9 times more efficient when $k = 10$.

5 Conclusion

In this paper, we have shown the possibility of a simple training-free framework for zero-shot text-to-image generation, and have examined two types of conditional GANs for our purpose. Our framework could also serve as a sanity check pipeline of conditional GANs for future works to more effectively leverage different kinds of GANs for more efficient search-based methods. While our method relies on pure random search in generating images, it is possible to explore more efficient methods based on discrete optimization, reinforcement learning or amortizing the latent code generator.

References

[1] 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, 2021. URL `https://arxiv.org/abs/2103.00020`.

[2] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. URL `https://arxiv.org/abs/2204.06125`.

[3] Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. Towards language-free training for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17907–17917, June 2022.

[4] Federico Galatolo., Mario Cimino., and Gigliola Vaglini. Generating images from caption and vice versa via clip-guided generative latent space search. *Proceedings of the International Conference on Image Processing and Vision Engineering*, 2021. doi: 10.5220/0010503701660174.

[5] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 88–105, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19836-6.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[7] C Wah, S Branson, P Welinder, P Perona, and S Belongie. Technical report cns-tr-2011-001. *California Institute of Technology: Pasadena, CA, USA*, 2011.

[8] Dong Huk Park, Samaneh Azadi, Xihui Liu, Trevor Darrell, and Anna Rohrbach. Benchmark for compositional text-to-image synthesis. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[9] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Finding an unsupervised image segmenter in each of your deep generative models. 2021.

[10] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks, 2019.

[11] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis, 2019.

A Detailed Best CLIP Score and FID

Method	k=1	k=10	k=20	k=50	k=100	k=1000
SAGAN seed=42	0.298575	0.311092	0.310343	0.306805	0.302977	0.288241
SAGAN seed=43	0.298602	0.311327	0.310399	0.306534	0.302771	0.287837
SAGAN seed=44	0.298783	0.311415	0.310338	0.306734	0.302756	0.288657
AVG	0.298653	0.311278	0.31036	0.306691	0.302835	0.288245
STD	$1.131E-4$	$1.669E-4$	$3.387E-5$	$1.405E-4$	$1.235E-4$	$4.1E-4$
BIGGAN seed=42	0.279637	0.304543	0.305433	0.302779	0.298637	0.28176
BIGGAN seed=43	0.279755	0.304347	0.305115	0.302454	0.298424	0.281907
BIGGAN seed=44	0.279842	0.304496	0.305286	0.302718	0.298877	0.281914
AVG	0.279745	0.304462	0.305278	0.30265	0.298646	0.281860
STD	$1.029E-4$	$1.023E-4$	$1.591E-4$	$1.724E-4$	$2.266E-4$	$8.696E-5$

Method	k=1	k=10	k=20	k=50	k=100	k=1000
SAGAN seed=42	107.21	60.98	56.64	56.20	57.77	77.65
SAGAN seed=43	105.58	59.89	55.53	56.76	56.48	75.54
SAGAN seed=44	106.32	60.69	54.7	55.46	57.08	74.93
AVG	106.37	60.52	55.62	56.14	57.11	76.04
STD	0.82	0.56	0.97	0.65	0.65	1.43
BIGGAN seed=42	117.63	95.81	92.91	90.35	86.58	84.69
BIGGAN seed=43	117.57	95.75	94.07	90.43	86.96	83.22
BIGGAN seed=44	117.32	96.12	93.59	91.2	88.6	84.68
AVG	117.51	95.89	93.52	90.66	87.38	84.20
STD	0.1644	0.1986	0.5829	0.4694	1.0735	0.8458

Table 3: Detailed Best CLIP (upper) score and FID (below)

B Examples of Generated Image

We show some example images generated by our proposed method in Figure. 3. All images are selected from 100 random generations with the highest CLIP score, expect that the ground truth images are from the dataset. We can see the generated images are more relevant to the text prompt when we only use the top-10 ImageNet labels than purely random generation (k=1000).

C Contributions of Each Group Member

Kejia Yin proposed to select TOP-K ImageNet labels to improve generation efficiency. Implemented the proposed method based on some initial code from Xuduo Gu. Conducted experiments with SAGAN and BIGGAN. Composed the Experiment section of the final report.

Xuduo Gu implemented the original version of the model, which was later modified by Kejia Yin for the final experiment. Proposed heuristic search for z (deprecated). Composed the Abstract, Method, and some of the Related Work and Experiment sections of the final report.

Linfeng Du helped with the framework design and implementation, and implemented the quantitative evaluation metrics for GANs (CLIP score and Fréchet Inception Distance). Composed the introduction and conclusion sections of the final report.

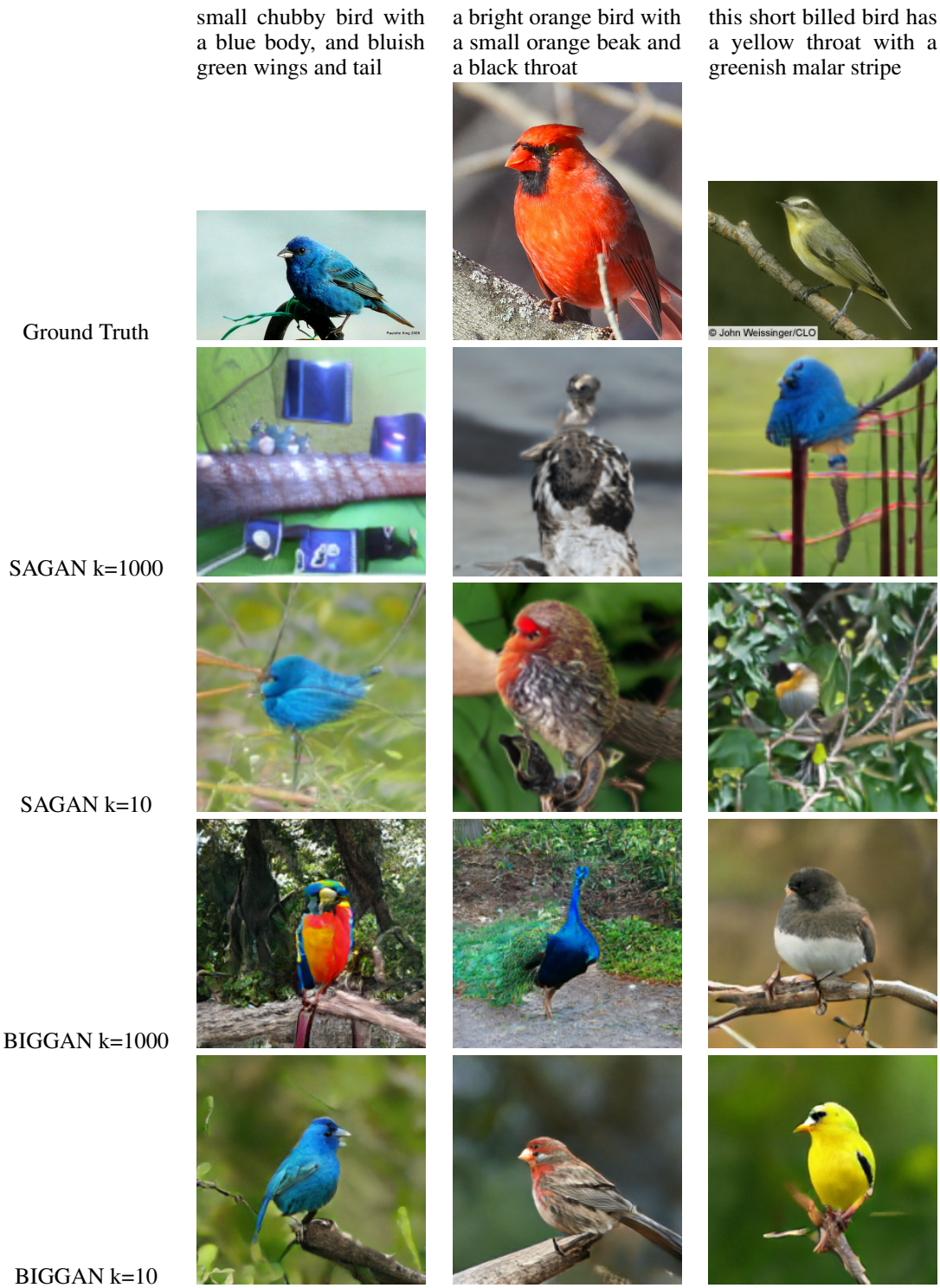

Figure 3: Some examples of images generated by proposed method