FIGGEN: TEXT TO SCIENTIFIC FIGURE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The generative modeling landscape has experienced tremendous growth in recent years, particularly in generating natural images and art. Recent techniques have shown impressive potential in creating complex visual compositions while delivering impressive realism and quality. However, state-of-the-art methods have been focusing on the narrow domain of natural images, while other distributions remain unexplored. In this paper, we introduce the problem of text-to-figure generation, that is creating scientific figures of papers from text descriptions. We present FigGen, a diffusion-based approach for text-to-figure as well as the main challenges of the proposed task. Code and models will be made public.

1 INTRODUCTION

Scientific figure generation is an important aspect of research, as it helps to communicate findings in a concise and accessible way. The automatic generation of figures presents numerous advantages for researchers, such as savings in time and effort by utilizing the generated figures as a starting point, instead of investing resources in designing figures from scratch. Making visually appealing and understandable diagrams would allow accessibility for a wider audience. Furthermore, exploring the generative capabilities of models in the domain of discrete graphics would be of high interest.

Generating figures can be a challenging task, as it involves representing complex relationships between discrete components such as boxes, arrows, and text, to name a few. Unlike natural images, concepts inside figures may have diverse representations and require a fine-grained understanding. For instance, generating a diagram of a neural network presents an ill-posed problem with high variance, as it can be represented by a simple box or an unfolded representation of its internal structure. Human understanding of figures largely relies on the text rendered within the image, as well as the support of text explanations from the paper written in technical language.

By training a generative model on a large dataset of paper-figure pairs, we aim to capture the relationships between the components of a figure and the corresponding text in the paper. Dealing with variable lengths and highly technical text descriptions, different diagram styles, image aspect ratios, and text rendering fonts, sizes, and orientations are some of the challenges of this problem. Inspired by impressive results in text-to-image, we explore diffusion models to generate scientific figures. Our contributions are i) introduce the task of text-to-figure generation and ii) propose FigGen, a latent diffusion model that generates scientific figures from text captions.

Related work. Deep learning has emerged as a powerful tool for conditional image generation (Ramesh et al., 2022; Saharia et al., 2022; Balaji et al., 2022), thanks to advances in techniques such as GANs (Goodfellow et al., 2014; Karras et al., 2019; 2021) and Diffusion (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020b). In the domain of scientific figures, Rodriguez et al. (2023) presented Paper2Fig100k, a large dataset of paper-figure pairs. In this work, we aim to explore diffusion models applied to the task of text-to-figure generation and analyze its challenges.

2 METHOD AND EXPERIMENTS

We train a latent diffusion model (Rombach et al., 2021) from scratch. First, we learn an image autoencoder that projects images into a compressed latent representation. The image encoder uses a KL loss and OCR perceptual loss (Rodriguez et al., 2023). The text encoder used for conditioning is learned end-to-end during the training of the diffusion model. The diffusion model interacts directly in the latent space and performs a forward schedule of data corruption while simultaneously learning

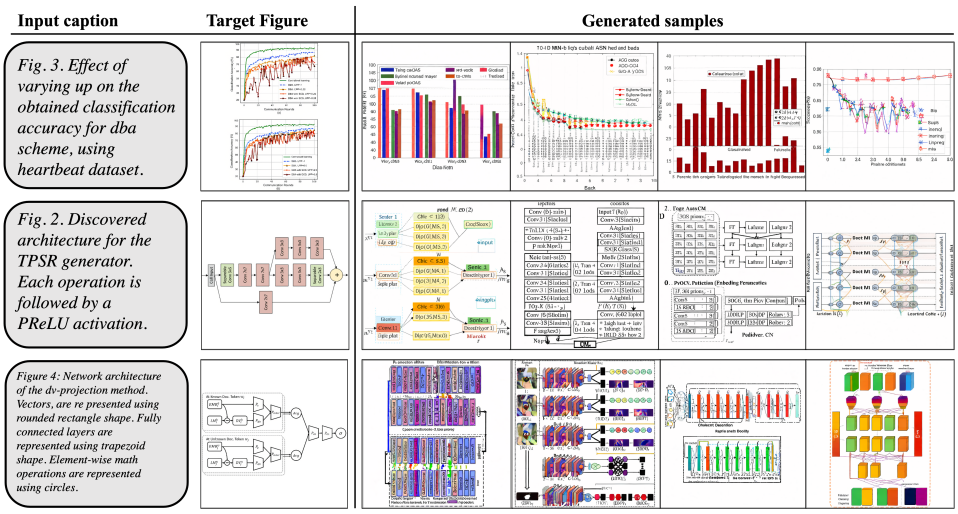

Figure 1: Samples generated by our model using captions from Paper2Fig100k test set.

Model	Text encoder	Parameters	CFG	FID↓	ISC↑	KID↓
$\text{FigGen}_{\text{Base}}$	Bert (8 layers)	866M	1.0	302.46	1.04	0.32
$\text{FigGen}_{\text{Base}}$	Bert (8 layers)	866M	5.0	282.32	**1.09**	**0.29**
$\text{FigGen}_{\text{Base}}$	Bert (8 layers)	866M	10.0	284.12	1.08	**0.29**
$\text{FigGen}_{\text{Mid}}$	Bert (32 layers)	942M	1.0	308.50	1.03	0.32
$\text{FigGen}_{\text{Mid}}$	Bert (32 layers)	942M	5.0	298.98	1.06	0.31
$\text{FigGen}_{\text{Mid}}$	Bert (32 layers)	942M	10.0	301.10	1.06	0.31
$\text{FigGen}_{\text{Large}}$	Bert (128 layers)	1.2B	1.0	302.99	1.04	0.32
$\text{FigGen}_{\text{Large}}$	Bert (128 layers)	1.2B	5.0	**281.25**	**1.09**	**0.29**
$\text{FigGen}_{\text{Large}}$	Bert (128 layers)	1.2B	1.0	288.02	**1.09**	**0.29**

Table 1: Main quantitative results of our text to figure generation models.

to revert the process through a time and text conditional denoising U-Net (Ronneberger et al., 2015) (see Appendix A.1 for details). We use Paper2Fig100k, composed of figure-text pairs from research papers. It consists of $81,194$ samples for training and $21,259$ for validation.

Experimental results. During generation, we use DDIM (Song et al., 2020a) sampler with 200 steps and generate $12,000$ samples for each model to compute FID, IS, and KID (Regenwetter et al., 2023). We use classifier-free guidance (CFG) to test super-conditioning (Ho & Salimans, 2022). Table 1 presents results of different text encoders, and Figure 1 shows generated samples of $\text{FigGen}_{\text{Base}}$. We find that the small text encoder offers the best results and that we can improve conditional generation by increasing the CFG scale. Although qualitative samples do not present sufficient quality to solve the task, FigGen has learned interesting relationships between texts and figures such as the difference between plots and architectures (see also Appendix A.3).

3 Conclusion

In this paper, we introduce the task of text-to-figure generation and define FigGen, a latent diffusion model that we train on the Paper2Fig100k dataset. Our experiments show that FigGen is able to learn relationships between figures and texts and generate images that fit the distribution. However, these generations are not ready to be useful for researchers. One of the main challenges to solve is the variability in text and images, and how to better align both modalities. Also, future work must design validation metrics and loss functions for generative models of discrete objects.

Ethics statement. A central concern of this work is fake paper generation. To address this, we consider building classifiers or using watermarks for the detection of fake content. However, further research is needed to elucidate how these systems should be made public.

URM Statement. The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2023 Tiny Papers Track.

REFERENCES

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. *arXiv*, abs/2012.09841, 2020.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv*, abs/1406.2661, 2014.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *arXiv*, abs/1912.04958, 2019.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021.

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. *arXiv*, abs/2103.00020, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv*, abs/2204.06125, 2022.

Lyle Regenwetter, Akash Srivastava, Dan Gutfreund, and Faez Ahmed. Beyond statistical similarity: Rethinking metrics for deep generative models in engineering design. *arXiv preprint arXiv:2302.02913*, 2023.

Juan A Rodriguez, David Vazquez, Issam Laradji, Marco Pedersoli, and Pau Rodriguez. Ocr-vqgan: Taming text-within-image generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3689–3698, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *arXiv*, abs/2112.10752, 2021.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241. Springer, 2015.

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*, abs/2205.11487, 2022.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *arXiv*, abs/1503.03585, 2015.

Input (latent) shape	64 x 64 x 4
Number of channels	256
Number of residual blocks	3
Self-attention resolutions	[64, 32, 16]
Channel mult.	[1, 2, 4, 4]
Dropout	0

Table 2: Base diffusion U-Net architecture.

Input (image) shape	384 x 384 x 3
Embed dimension	4
Number of channels	128
Number of residual blocks	2
Channel mult.	[1, 2, 4, 4]
Discriminator weight	0.5
VGG perceptual loss weight	0.2
OCR perceptual loss weight	0.8
KL loss weight	1e-6
Dropout	0

Table 3: Image autoencoder architecture.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *arXiv*, abs/1801.03924, 2018.

A APPENDIX

A.1 MODEL DETAILS

Image encoder. The first stage, the image autoencoder, is devoted to learning a projection from the pixel space to a compressed latent representation that makes the diffusion model train faster. The image encoder needs to also learn to project the latents back to the pixel space without losing important details about the figure (e.g., text rendering quality). To this end, we define a convolutional encoder and decoder with a bottleneck, that downsamples images with a factor $f = 8$. The encoder is trained to minimize a KL loss with a gaussian distribution as well as a VGG perceptual (Zhang et al., 2018) loss and an OCR perceptual loss. We follow the adversarial procedure proposed in VQGAN (Esser et al., 2020), which increases reconstruction quality.

Text encoder. We find that using a general-purpose text encoder (e.g., CLIP (Radford et al., 2021)) is not well-suited for our task, because text encoders trained on natural texts and images exhibit a domain gap with respect to the technical descriptions present in papers. We define Bert (Devlin et al., 2018) transformer that is trained from scratch during the diffusion process. We use an embedding channel of size 512, which is also the embedding size of the cross-attention layers for conditioning the U-Net. We explore varying the number of layers of the transformer in the set $8, 32, and 128$.

Latent diffusion model. Table 2 describes the U-Net network architecture. We perform diffusion in a perceptually equivalent latent representation of images, that is compressed to an input size of $64x64x4$, which makes the diffusion model faster. We define 1000 steps of diffusion and a linear noise schedule.

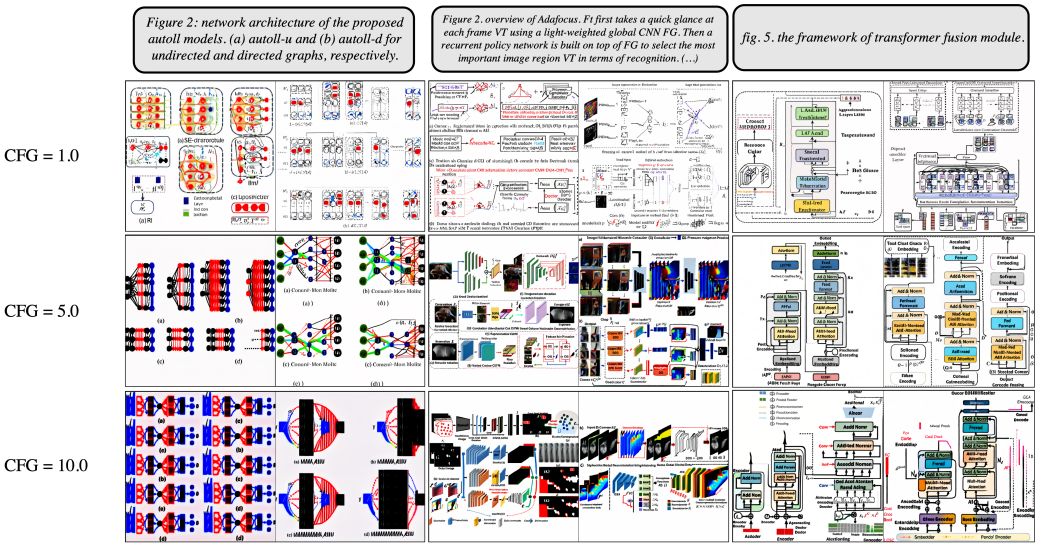

Figure 2: Samples generated from FigGen$_{\textbf{Base}}$. We display two generated samples for three prompts Paper2Fig100k test set. Each row displays the classifier-free guidance (CFG) scale, showing that we can super-condition the generations to make the samples align better with the prompt.

A.2 Training details

Our models are trained in Paper2Fig100k. During our experiments, we find that a challenging problem is how to deal with extremely varying aspect ratios that exist between images in the dataset (e.g., figures tend to be larger in width than in height). Cropping figures would result in a loss of crucial information. Therefore we opt for applying white padding to images and only consider images with an aspect ratio between 0.5 and 2, to avoid having most of the pixel information be the padded pixels. The final dataset consists of $37,613$ samples for training and $9,506$ for validation. Images are processed at size $512x512$ and downsampled to $64x64$ by our image autoencoder.

For training the image autoencoder we use Adam optimizer with an effective batch size of 4 samples, and a learning rate of $4.5e-6$, using 4 Nvidia V100 12GB GPUs. For training stability, we warm up the model during $50k$ iterations without using the discriminator (Esser et al., 2020; Rombach et al., 2021). For training latent diffusion models, we use Adam optimizer, using an effective batch size of 32 and a learning rate of $1e-4$. For training the models on Paper2Fig100k we use 8 Nvidia A100 80GB GPU.

A.3 Additional generation results

Figures 2 show additional generated samples of FigGen when tuning the classifier-free guidance (CFG) (Ho & Salimans, 2022) parameter. We observe improvement in figure quality when increasing the CFG scale which is also shown quantitatively. Figure 3 presents more generations of FigGen. Note the variability in text length between samples, as well as the technical level of the captions, which makes it difficult for the model to properly generate understandable figures. However, high-level concepts are correctly captured, such as captions that describe charts and plots, algorithms, or cases where we aim to display neural network architectures.

Figure 2. Architectures of inversion attack models. a) baseline threat model with target cnn model (…) confidences are input to a transposed cnn (tcnn) for inversion attack (d). b) threat model with explainable target model that (…)

Figure 2: The overview of our HDR reconstruction network architecture.

Figure 10: generator and discriminator architectures. (a) c is a learned input constant, z is the random noise vector input, and ni are noise maps that are fed to different layers. y is a one-hot vector containing the class information. a styleconv module consists of style modulation, convolution, and noise injection. fc represents fully-connected layers. (b) (…)

Figure 3: Our proposed architecture of extended cross-domain representation disentangler (e-cdrd), which jointly performs cross-domain representation disentanglement and image translation.

Figure 1: The duo classifier model architecture

Fig. 3. Framework for action sequencing using visual permutations.

Figure 1: The figure shows the complete architecture. here AFP minimizes the number of filter in model while prc maximizes the accuracy during pruning. here t,wt and fi are the regularization parameter, weight-threshold and remaining filters in the model respectively at tth pruning iteration.

Figure 2. Icarus architecture.

Figure 1: The architecture of self-gradient network. the input data is passed through the network once and the soft-loss is calculated based on (4). the computed soft loss value is fed into the self-gradient block which is added to the input data when it is processed and the main network predicts the output (…)

Figure 2. Schematic representation of the procedure followed to obtain the training sequences and the codebooks for a given user p.

Figure 1: System overview for annotating news articles and enabling structured search.

Fig. 2: Model overview. the encoder (left) first converts the 3d input x (e.g., noisy point clouds or coarse voxel grids) into features using task-specific neural networks. next, the features are projected onto one or multiple planes (fig. 2a) or into a volume (fig. 2b) using average pooling. the conv (…)

Fig. 11. The quantitative performances of the proposed method and its degraded versions, in which the is in eq. (3) are set as 1015 in turn.

Fig. 5. The direct sbs-serving ratio for algorithm 1, the mpcc scheme, the rl-nc scheme, and the rl-ucc scheme in the small-scale system.

Figure 3: the architecture of our proposed contextualized temporal attention mechanism. three stages are proposed to capture the content information at stage with selfattention, temporal information at (…)

Figure 3. The proposed neural network architecture.

Figure 3: Generated samples from the test set prompts above as input. Samples generated using FigGen$_{\textbf{Base}}$.