Adding Conditional Control to Text-to-Image Diffusion Models

Subhan Kamil
Technical University of Munich
subhan.kamilov@tum.de

Abstract

In this work, we present a report on ControlNet, a neural network architecture designed to enhance large text-to-image models with task-specific conditions. The authors of ControlNet investigate the challenges of large text-to-image diffusion models in learning conditioning associated with limited dataset sizes, computation constraints, and diverse image processing tasks. ControlNet addresses these challenges by modifying the neural network block and introducing Zero Convolution. The report highlights the main contribution of ControlNet and raises concerns about implementation details and comparisons with previous works. Overall, ControlNet provides a valuable approach to controlling large pre-trained models and offers insights for future research in the field.

1. Introduction and Motivation

Large text-to-image diffusion models allow for generating visually appealing images with short descriptive prompts. However, questions arise regarding the effectiveness of prompt-based control and more importantly, the applicability of these models to specific image-processing tasks. Thus, the authors of the "Adding Conditional Control to Text-to-Image Diffusion Models" (ControlNet) [20] performed a thorough investigation of various image processing applications and revealed three key findings:

1. Limited dataset sizes in task-specific domains, for example, normal maps as conditioning from [15] (less than 100k data samples), require robust neural network training methods to avoid overfitting and preserve the generalization ability of the large text-to-image models like [12] trained on 5B LAION dataset [13].

2. Commonly, lack of access to large computation clusters emphasizes the need for fast training methods that optimize large models within limited time and memory constraints. In turn, this leads to the necessity of the use of pre-trained weights, fine-tuning, and transfer learning.

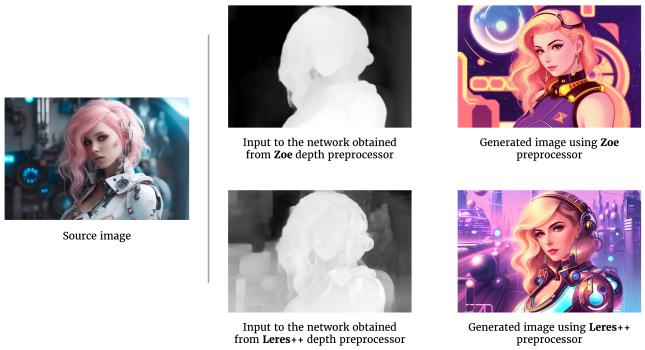

Figure 1. Example of using ControlNet to condition the generation on depth information. Images were generated using "**a woman retrofuturism**" as a prompt [2].

3. Since some image processing problems (e.g. the pose-to-human task) have diverse forms of problem definitions, user controls, and image annotations, they require end-to-end learning to interpret raw inputs into object-level or scene-level understandings.

Motivated by this, the authors of [20] propose a new neural network structure, named ControlNet, which can enhance large pre-trained text-to-image diffusion models with task-specific conditions.

2. Approach

The main contribution of the paper [20] is the introduction of a neural network architecture to add additional control to the pre-trained text-to-image models. Specifically, this method was applied and tested on the pre-trained neural network from Stability AI, weights of which were released to the public [1].

Before introducing ControlNet, it is important to reiterate the definition of the ordinary neural network block. As "ordinary neural network block" authors of ControlNet [20] refer to a set of neural layers, a combination of which make up a block, which is a frequently used unit to build neural networks (e.g. "resnet" block, "conv-relu-bn" block). In its principle, ordinary neural network block takes an input feature map $\boldsymbol{x}$, which in case of 2D will be $\boldsymbol{x} \in \mathbb{R}^{h \times w \times c}$ (with

$\{h, w, c\}$ denoting height, width, and channels) passes it through neural layers in its block and transforms it into another feature map $\boldsymbol{y}$.

To form a ControlNet block, authors propose to lock the ordinary neural network block i.e. freeze its parameters, and create a trainable copy of it. Then, given $\boldsymbol{c}$ that is an external condition that we want to add to the neural network, we pass it through a unique type of connection layer called Zero Convolution, next we add the output from zero convolution to the original feature map from diffusion model and pass it to the trainable copy. After the trainable copy, the resulting transformation is again passed through another zero convolution layer and added to the output from the original "ordinary neural network block", resulting in new, conditioned output y_c. See Fig.2 for a schematic depiction of the ControlNet block compared to the ordinary neural network block.

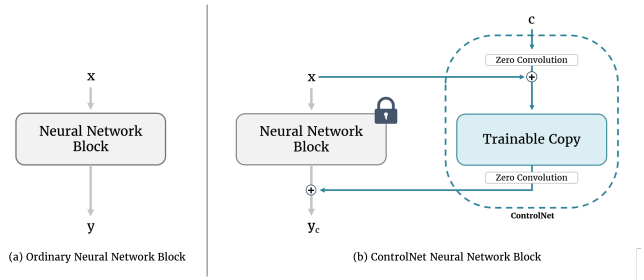

Figure 2. ControlNet structure applied on a single neural network block.

Zero Convolution in this case refers to the convolutional layer of kernel size 1×1 with both weights and bias initialized as zero.

Authors of ControlNet motivate specific design choices with the following:

- Division of the network into locked and trainable copies helps to avoid overfitting when the dataset is small and to preserve the production-ready quality of large pre-trained models. Specifically, the locked copy preserves the network capability learned from billions of images, while the trainable copy is trained on task-specific datasets to learn the conditional control.

- Zero Convolution allows the weights to progressively grow from zeros to optimized parameters in a learned manner, thus, they do not add new noise to deep features of the large pre-trained network. This also results in the training as fast as fine-tuning a diffusion model, compared to training new layers from scratch.

An example of the ControlNet applied on the Stable Diffusion model based on [12] can be seen in Fig.3. The U-net structure used in Stable Diffusion consists of 25 neural network blocks, 12 of which are encoder, 1 middle, and 12 decoder blocks [12]. In particular, authors of ControlNet [20] created trainable copies of 12 encoder and 1 middle blocks of Stable Diffusion, outputs of which were added to the 12 skip-connections to the decoder and 1 middle block of the U-net.

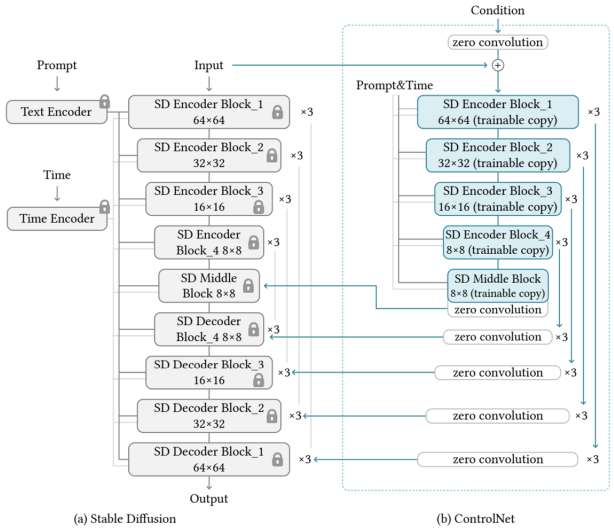

Figure 3. ControlNet in Stable Diffusion. The gray blocks are the original neural network blocks of the Stable Diffusion and blue box denote ControlNet blocks.

The ControlNet architecture uses a similar learning objective as most of the diffusion models, i.e. given a set of conditions like time-step, text-prompts, and task-specific conditions, the model learns weights to predict the noise added to the image. For a detailed description we refer to the original paper of authors [20], section 3.3.

3. Main Results

Initially, the authors presented eight implementations of ControlNet with various image-based conditions. In this work, we show examples of the results of the ControlNet trained on three conditions (see Fig. 4) and refer to the original paper [20] for examples from other conditioning tasks and extensive generated images in the appendix. It is worth mentioning that those authors defined three ways of using prompts to test the generated images:

1. Default Prompt. For the default prompt "a professional, detailed, high-quality image" was used.

2. Automatic Prompt. A prompt that was generated from image captioning methods, e.g. BLIP [11].

3. User Prompt. Classic prompt provided by the user.

Furthermore, the authors sampled datasets of different scales from the Canny Edge dataset and tested the generation capability of the network based on various dataset sizes. It has been noted that with only 10k data samples ControlNet concept can generate plausible results (see Fig. 5 (a)). Apart from that, the authors emphasize the ”sudden convergence phenomenon”, when the model starts to follow the input instructions (Fig. 5 (b)). This phenomenon was observed during training from 5000 to 10000 steps, and described as a consequence of using zero convolution operation in the network (i.e. gradually affecting deep features of the large model).

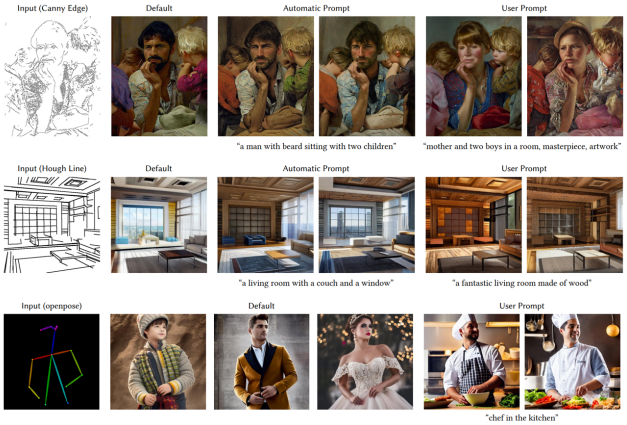

Figure 4. Example of ControlNet architecture trained on three different conditions.

In the original paper, only one limitation of the proposed method is mentioned, which is the challenge of eliminating the semantic bias of unconditional generated image in the generated image using prompts. See Fig. 28 in the original paper [20] for details.

4. Related Work

Related work section in the original work [20] begins with the mentioning of the earliest paper on Hypernetworks [6]. We, however, think that the main motivation of the work came from the hypernetworks introduced by Kurumuz in the respective blog post by NovelAI [9]. In [9] authors explicitly mention the difference between the earliest work on hypernetworks [6] that uses separate neural network to modify or generate the weights of the main neural network, while hypernetworks from [9] work by applying a single small neural network at multiple points within the larger network, modifying the hidden states. In particular, hypernetworks were applied in the transformer block of the Stable Diffusion. Later this approach has been found to perform well and sometimes even better than finetuning large models. Moreover, this method excelled particularly when data were scarce on the desired task, producing better results than fine-tuning the large model [9].

The hypernetworks concept introduced by NovelAI was further improved by independent contributors in various discussion forums and gained popularity after [7] shared hyperparameters and training tricks for getting good results from hypernetworks.

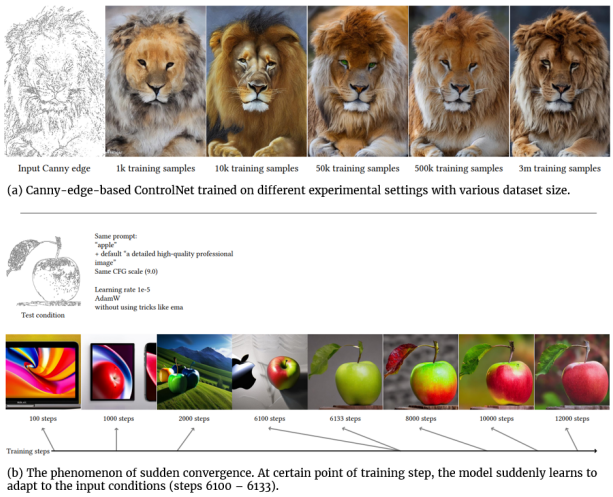

Figure 5. Experiment of training on datasets of various sizes (a). The sudden convergence phenomenon during training (b).

ControlNet [20], modifies the idea of hypernetworks discussed in [9], and instead of only adding small neural networks in the transformer part of the U-net architecture, they add it in different parts of the U-net structure before the input feature map and after the produced feature map by the trainable neural network block.

Recently, an updated version, ControlNet 1.1, was published [19]. The new version preserved the same architecture while improving the generation quality and robustness of the previous conditioned models. Furthermore, several new conditioned models were added to the existing ones, like controlnet trained on Instruct Pix2Pix dataset [3] for instruction based image editing, Multi-ControlNet that allows multiple input conditions for a single generation, etc.

Some newly published works also take advantage of ControlNet in their work. In [18], for example, it is used to add spatial control of the subject in text-guided video-to-video translation. Visual ChatGPT - a system that incorporates ControlNet with ChatGPT to combine image understanding and controlling capabilities of ControlNet with the language understanding of ChatGPT to enable it to work not only with text data but also images; processing complex visual questions or instructions for visual editing that require utilization of capabilities of several models [17]. Furthermore, depth map conditioned ControlNet has also been

adopted for the task of text-to-3D generation, via providing a 3D depth prior for the generative model [14].

5. Personal Remarks

In our opinion, the main takeaway from this paper is the introduction of the general concept to control large pre-trained models via some additional conditions. Therefore, the idea of the ControlNet [20] should be taken into account not only with large text-to-image diffusion models but also in terms of other large models to influence their behavior without distorting the original weights.

Moreover, the fast pace of the field results in the lack of proper research on claimed quality and properties of the models. There are more and more contributions from many independent authors in various discussion forums and blogs without proper analysis of their respective publications. All of this results in many unanswered questions. For example, in the ControlNet paper, several implementation details have no proper explanation:

- Consider examples of three different ControlNet conditions shown in Figure 4. According to the authors [20] trained models have different dataset sizes and different training times for each kind of condition (see Fig. 6). This leads to the question of why we have a certain number of hours of training for one condition but not for the other. Or what is the reason for training to be stopped at this particular time; did ControlNet start overfitting at these timesteps?

- In Fig. 5 (a) authors show the generation results based on training on datasets of different sizes. But how much should we train to achieve plausible results without the risk of overfitting is not described, nor training times for different dataset sizes are given.

- Another concern is the unfair comparison of the results to the previous works. For example, in Fig. 18 in the original paper [20] authors compare ControlNet generation results to the Sketch-guided diffusion [16]. While authors of ControlNet emphasize qualitative performance, they fail to mention different training settings of the two models, i.e. ControlNet on a dataset of size 500k for 300 hours, whereas Sketch-guided diffusion claims to train only on a few thousand images for 1 hour [16].

- Next, it is not clear why should one train a ControlNet instead of finetuning the weights of the existing model if large datasets are available (e.g. Canny edge conditioning in [20] with 3M available data using Canny edge detector [4]). Since the ControlNet introduces additional compute demand from weights of zero convolutions apart from the original model weights in the trainable copy blocks. Overall, there is not much work comparing both approaches yet to claim which one is superior.

- Finally, the original paper lacks quantitative assessments of the generated images, for example, Fréchet Inception Distance (FID) introduced in [8] for measuring the overall quality; precision and recall metrics for measuring the fidelity and diversity of the generated images [10]. Thus, it adds additional complexity of comparing it with similar methods, e.g. Sketch-guided diffusion [16].

Figure 6. Training hours with respective dataset sizes of the examples presented in Figure 4. All models were trained on NVIDIA A100 80G.

In conclusion, since there is a lack of research fairly comparing the results of ControlNet to plain finetuning of the existing Stable Diffusion network or to different conditioning approaches (e.g. [3, 16, 5]). We assume that the popularity of ControlNet can be attributed to: publishing 8 conditionally trained models right away, while others works finetune only on 1 or 2 input conditions; previous popularity of the author from the paper in a related domain [21] that resulted in similar recognition of the new work.

6. Summary

To summarise, in this report, we introduced the ControlNet from [20], as a general concept that can be used to enhance large text-to-image models with task-specific conditions. Although the original paper presented ControlNet to influence the generation results of the Stable Diffusion model, we believe the application of ControlNet is not limited only to it. The report highlights the approach and main contribution of the ControlNet and raises concerns about implementation details and comparisons with similar works.

7. Acknowledgements

We would like to thank Dr. Andreas Rössler for organising this interesting seminar and for fruitful discussions and insights along the course.

References

[1] Stability AI. Stable diffusion public release, 2022. https://stability.ai/blog/stable-diffusion-public-release.

[2] Stable Diffusion Art. Controlnet generations using depth maps as conditioning. https://shorturl.at/hipxM.

[3] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions, 2023.

[4] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[5] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis, 2021.

[6] David Ha, Andrew Dai, and Quoc V. Le. Hypernetworks, 2016.

[7] Heathen. Hypernetwork style training, a tiny guide, 2022. github.com/automatic1111/stable-diffusion-webui/discussions/2670.

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017.

[9] Kurumuz. Novelai improvements on stable diffusion, 2022. https://blog.novelai.net/novelai-improvements-on-stable-diffusion-e10d38db82ac.

[10] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models, 2019.

[11] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022.

[12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[13] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022.

[14] Junyoung Seo, Wooseok Jang, Min-Seop Kwak, Jaehoon Ko, Hyeonsu Kim, Junho Kim, Jin-Hwa Kim, Jiyoung Lee, and Seungryong Kim. Let 2d diffusion model know 3d-consistency for robust text-to-3d generation, 2023.

[15] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R. Walter, and Gregory Shakhnarovich. Diode: A dense indoor and outdoor depth dataset, 2019.

[16] Andrey Voynov, Kfir Aberman, and Daniel Cohen-Or. Sketch-guided text-to-image diffusion models, 2022.

[17] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models, 2023.

[18] Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation, 2023.

[19] Lvmin Zhang. Controlnet 1.1, 2023. https://github.com/lllyasviel/ControlNet-v1-1-nightly.

[20] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

[21] Lvmin Zhang, Yi Ji, and Xin Lin. Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan, 2017.