On the Lower Limit of Monocular Depth Diversity Required for ControlNet-Based Conditional Image Generation

Alexander Rosen
a.rosen@mail.utoronto.ca

Abstract

Recent advances in diffusion models have demonstrated impressive capabilities in conditional image generation, particularly when guided by structured modalities such as depth maps. In this work, we explore the limits of conditioning diffusion models using monocular depth maps without relying on extensive scene diversity in training data. Building on the ControlNet framework, we compare generation quality using two datasets: the diverse SUN RGB-D and a custom-collected, environment-specific dataset from the Bahen Centre for Information Technology using an Azure Kinect DK ToF camera. We introduce a robust depth preprocessing pipeline—including non-local means filtering—to normalize dataset quality and ensure a fair comparison. Through empirical evaluation using FID, LPIPS, and a depth adherence metric, we characterize the lower bounds of scene diversity necessary for effective depth-based conditioning. Our findings suggest that even with constrained scene variety, conditional diffusion can yield high-fidelity, depth-consistent generations, paving the way for more accessible depth-conditioned generative systems.

Figure 1. Results taken from the ControlNet paper for conditioning on Canny edge detections

1. Introduction

1.1. Problem

Diffusion models [29] [31] [13] [32] have emerged as qualitatively and quantitatively successful methods for training generative models by predicting noise we add to a training sample through a series of discrete steps, or a *score function* [15], the gradient of log-likelihoods at continuous timesteps. This training procedure gave way to simple extensions that facilitate precise yet diverse conditional sampling; we can train conditional and unconditional models with the goal of fusing samples from both models at inference time during denoising, thereby creating a strong conditioning mechanism while preventing mode collapse.

Although these models tend to be used for conditioning on textual information, the authors of ControlNet [38] show impressive results for conditioning on several text and image modalities. One example, shown in Figure 1, is the result of conditioning on Canny edge detections [5], which functions as an excellent method to sample images guided by input human-drawn sketches. There are also results for conditioning on data that is much more expensive to obtain, such as depth map and image pairs.

1.2. Our hypothesis

The relationship between images and depth maps is an ambiguous one. That is, given an image, one can produce several reasonable depth maps, and vice versa. For most applications, this is obviously problematic, but we postulate that it may work to our advantage in the case of conditional generative modeling; it should be possible to get a general sense of depth without image and depth map pairs from a wide variety of environments (roads, parks, living rooms, etc.). We find this likely since, although a model might not

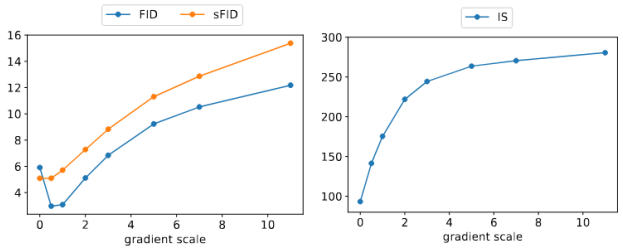

Figure 2. FID, sFID, and IS as a function of gradient scale w (taken from the classifier-free guidance paper)

see a depth map of a car, a clever enough sampler should be able to extrapolate a reasonable image from a conditional model's notation of depth if it is grounded by an unconditional model trained on plenty of car images. So, in this work, we provide a foundation for empirically characterizing a lower bound on the data diversity needed to condition ControlNet on monocular depth maps represented in image form.

2. Related Work

2.1. Conditional Generation

ControlNet was not the first paper to introduce the idea of conditing image generation on depth maps. Early on there were several attempts at learning conditional image gernation, including, but not limited to, Atribute2Image [36] which explored conditioning varational autoencoders (VAEs) [18] on visual attributes, PixelCCN [33] for image generation given a few words, and models such as StarGAN [6] or ContraGAN [17], generative adversarial networks (GANs) [10] for image manipulation or full generation given one-hot-encoded lables. While conditional image generation remained a challenging task for other generative modeling frameworks, diffusion models exploded in popularity, in large part due to the successes of classifier and classifier-free guidance [12] [8].

2.2. Diffusion-Based Conditional Generation

Some of the original diffusion papers, such as Denoising Diffusion Probabilistic Models [14], included results for control over color variation and image inpainting. Further work included conditioning on text such as CLIP latents [23] [22], conditioning on segmentation masks [9] [1], subject-driven generation [26], conditioning on sketches [34], and more general approaches aimed at a wide variety conditional data [2].

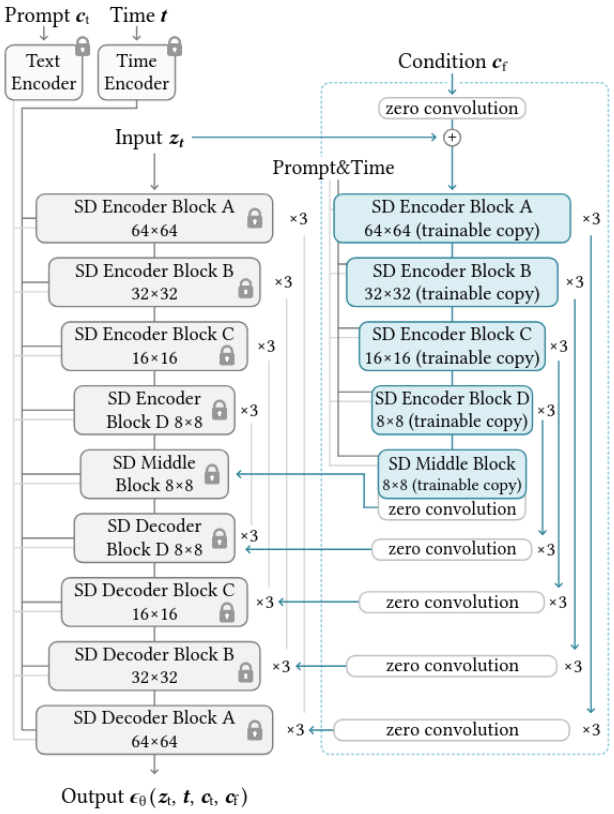

Figure 3. ControlNet architecture with a Stable Diffusion U-Net backbone (taken from the ControlNet paper)

3. Background

3.1. Denoising Diffusion Probabilistic Models

The formulation of diffusion we focus on is Denoising Diffusion Probabilistic Models (DDPMs) [14], which belong to a class of generative models whose sampling process involves moving backwards through a finite first-order Markov chain starting from an equilibrium that is tractable to sample from. The forward process is defined as

$$q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbb{I}) \tag{1}$$

for which we also have direct sampling given x_0:

$$q(x_t|x_0) = \mathcal{N}(\sqrt{\overline{\alpha}_t}x_0, (1-\overline{\alpha}_t)\mathbb{I}) \tag{2}$$

with $\alpha_t = 1 - \beta_t$ and $\overline{\alpha}_t = \prod_{i=1}^{t} \alpha_t$. This process is repeated for T steps, and we choose $\{\beta_t\}_{t=1}^{T}$ so that equilibrium x_T is approximately element-wise standard normal. Using (2) with the reparameterization trick, we can write

$$x_0 = \frac{1}{\sqrt{\overline{\alpha}_t}}\left(x_t - \sqrt{1-\overline{\alpha}_t}\epsilon\right) \tag{3}$$

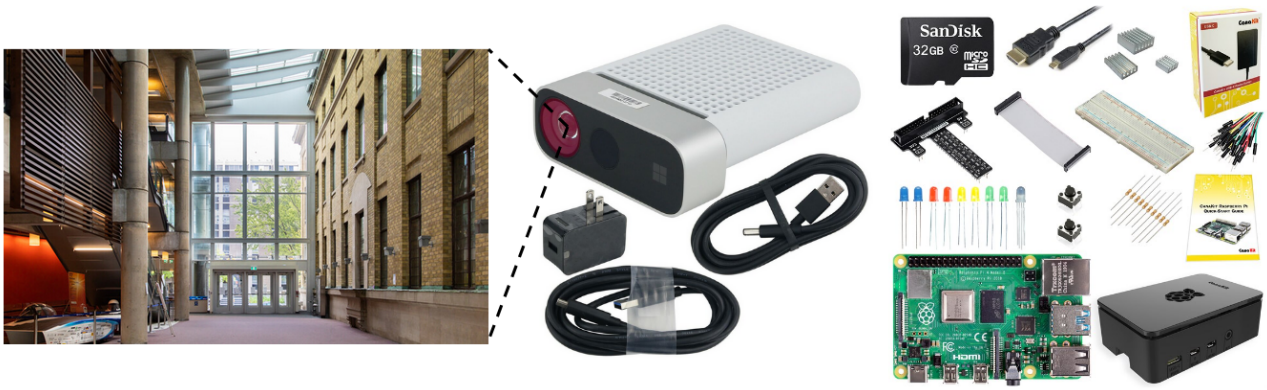

Figure 4. Hardware setup: left - Bahen, middle - Azure Kinect DK, right - Canakit Raspberry Pi 4

for $\epsilon \sim N(0, \mathbb{I})$, helping us show the mean of the reverse posterior $q(x_{t-1}|x_t, x_0)$ is

$$\mu(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right) \tag{4}$$

So, we train a neural network to predict

$$\epsilon_\theta(x_t, t) \approx \epsilon \tag{5}$$

and repeatedly sample x_{t-1} as μ (4) with varying amounts of additive Gaussian noise $\sigma_t z$ until we reach x_0:

$$x_{t-1} = \mu(x_t, t) + \sigma_t z \tag{6}$$

With the success of this method came many subsequent alterations, such as reasoning about more suitable forward coefficients [21], accelerating the backward process [30], and performing the diffusion process in a latent space [24].

3.2. Classifier-Free Guidance

Conditioning. Classifier-free guidance (CFG) [12] builds on classifier guidance [8], a method to train conditional diffusion models using a separate classifier trained at the same time. In classifier-free guidance, we train a diffusion model to always condition on label y, as in $\epsilon_\theta(x_t, t, y)$, but we randomly drop this label, replacing it as some null label $y = \varnothing$; this gives us one model that can be sampled from both conditionally and unconditionally. Now, by Bayes' rule,

$$\nabla_{x_t} \log p(x_{t-1}|x_t, y) = \nabla_{x_t} \log(x_t|x_{t-1}) + \nabla_{x_t} \log(y|x_{t-1}) \tag{7}$$

which we can use to show we need to add the gradient of some classifier's log probability to (4) to achieve conditional sampling:

$$\mu_{\text{cond}}(x_t, t) = \mu(x_t, t) + \sigma_t^2 \nabla_{x_t} \log p(y|x_t) \tag{8}$$

Since the score function $\nabla_{x_t} \log q(x_t)$ is equal to $-\epsilon_\theta(x_t, t)/\sqrt{1-\hat{\alpha}_t}$ [14], we can substitute this value into (7) to yield

$$\nabla_{x_t} \log p(x_{t-1}|x_t, y) = -\frac{\epsilon_\theta(x_t, t, y) - \epsilon_\theta(x_t, t, \varnothing)}{\sqrt{1-\hat{\alpha}_t}} \tag{9}$$

At sampling time, we use this identity (9) to end up with

$$\begin{aligned}\epsilon_{\text{cond}}(x_t, t, y) &= \epsilon_\theta(x_t, t, y) - \sqrt{1-\hat{\alpha}_t}\nabla_{x_t} \log(y|x_{t-1}) \\ &= 2\epsilon_\theta(x_t, t, y) - \epsilon_\theta(x_t, t, \varnothing)\end{aligned} \tag{10}$$

Controlling the amount of conditioning. We can take a barycentric combination of the conditional and unconditional score functions to get

$$\begin{aligned}\epsilon_{\text{guided}}(x_t, t, y) &= \epsilon_\theta(x_t, t, y) - w\sqrt{1-\hat{\alpha}_t}\nabla_{x_t} \log(y|x_{t-1}) \\ &= (w+1)\epsilon_\theta(x_t, t, y) - w\epsilon_\theta(x_t, t, \varnothing)\end{aligned} \tag{11}$$

where w controls the level of conditioning; $w = 0$ being unconditional, $w = 1$ being conditional (matching (10)), partial conditioning in-between, and $w > 1$ "super-conditioning". Figure 2 shows how varying w changes FID scores [11], sFID scores [20], and inception scores [27] (even though we should care more about FID [3]).

3.3. ControlNet

Let $y = \mathcal{F}_\theta(x)$ be the output of a neural network (or block) $\mathcal{F}$ with input x, such as $\mathcal{F}_\theta = \epsilon_\theta$ and $x = (x_t, t)$ in the case of (5). To transition to training a conditional generation model given a pre-trained unconditional model $\mathcal{F}$, we clone the parameters θ into θ_1, θ_2 where θ_1 is frozen and θ_2 we fine-tune. Call the output conditioned on c

$$y_c = \mathcal{F}_{\theta_1}(x) + \mathcal{F}_{\theta_2}(x + c) \tag{12}$$

Since this setup can be highly sensitive to noise for the first few gradient descent steps, we instead use

$$y_c = \mathcal{F}_{\theta_1}(x) + \mathcal{Z}_{\theta_{z2}}(\mathcal{F}_{\theta_2}(x + \mathcal{Z}_{\theta_{z1}}(c))) \tag{13}$$

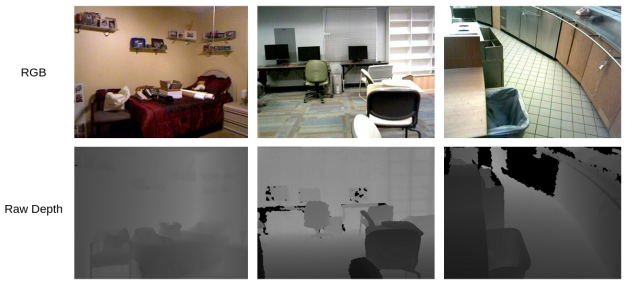

Figure 5. Raw SUN RGB-D pairs

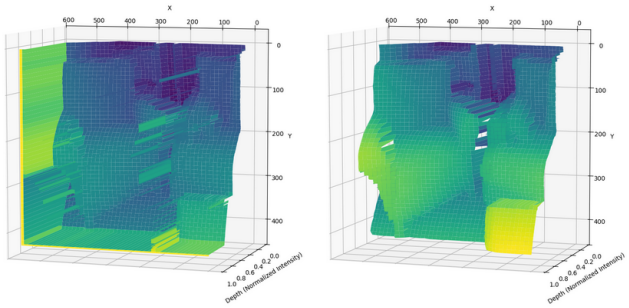

Figure 6. SUN RGB-D thresholding

Metric	SUN RGB-D	Bahen
completeness	84.4895	99.9898
depth range	0.2974	0.3765
entropy	4.0826	3.6904
noise level	0.0082	0.0051
size (# images)	10334	10020

Table 1. Comparison of depth map quality metrics between raw SUN RGB-D and our raw Bahen dataset

where $\mathcal{Z}$ are zero convolutions, 1×1 convolutions with weights and biases initialized to zero; $\theta_{z1}, \theta_{z2} = 0$. Intuitively, this setup leads to more stable initial training dynamics than (12) since we begin with $y_c = y$ without losing rich feature extraction given by θ_2. Lastly, instead of applying this augmentation to an entire network, we apply it to blocks of a network, such as each up and down block in a Stable Diffusion [24] U-Net [25], shown in Figure 3.

3.4. Non-Local Means

The Non-Local Means (NLM) algorithm [4] represents a significant departure from traditional local neighborhood filtering approaches for image denoising. Unlike conventional methods that operate on spatially adjacent pixels, NLM exploits the inherent redundancy in natural images by leveraging similarities between patches throughout the entire image domain, which is incredibly useful for our depth map processing pipeline. Given a noisy image v, the NLM algorithm estimates the denoised value at pixel i as a weighted average

$$\text{NLM}(v)(i) = \sum_{j \in I} w(i,j) v(j) \tag{14}$$

where weights $w(i,j)$ measure the similarity between patches centered at pixels i and j, defined as:

$$w(i,j) = \frac{1}{Z(i)} \exp\left(-\frac{\|P_i - P_j\|_{2,a}^2}{h^2}\right) \tag{15}$$

Here, P_i and P_j represent patches centered at pixels i and j respectively, $\|\cdot\|_{2,a}$ denotes a Gaussian-weighted Euclidean distance, h is a filter strength parameter controlling the decay of the exponential function, and $Z(i)$ is a normalizing constant ensuring $\sum_j w(i,j) = 1$.

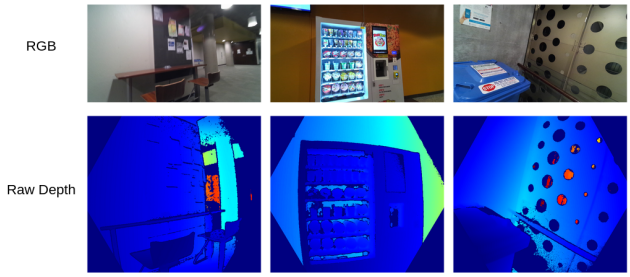

Figure 7. Raw Bahen dataset pairs

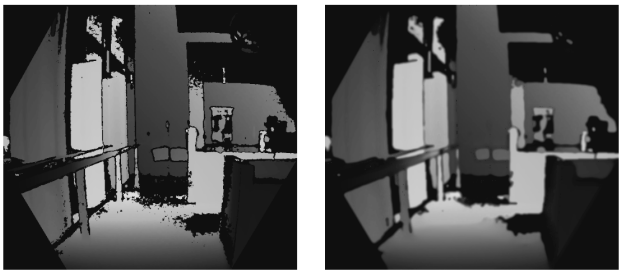

Figure 8. Bahen dataset non-local means

4. Experiments

4.1. Overview

For our experiments, since we want to compare generation for training data from diverse scenes versus limited scenes, we chose to use one off-the-shelf diverse dataset and collect our own semi-scene-specific one. For the diverse dataset, we selected the SUN RGB-D dataset from Princeton [40], which contains RGB-d images from NYU depth v2 [28], Berkeley B3DO [16], and SUN3D [35]. For

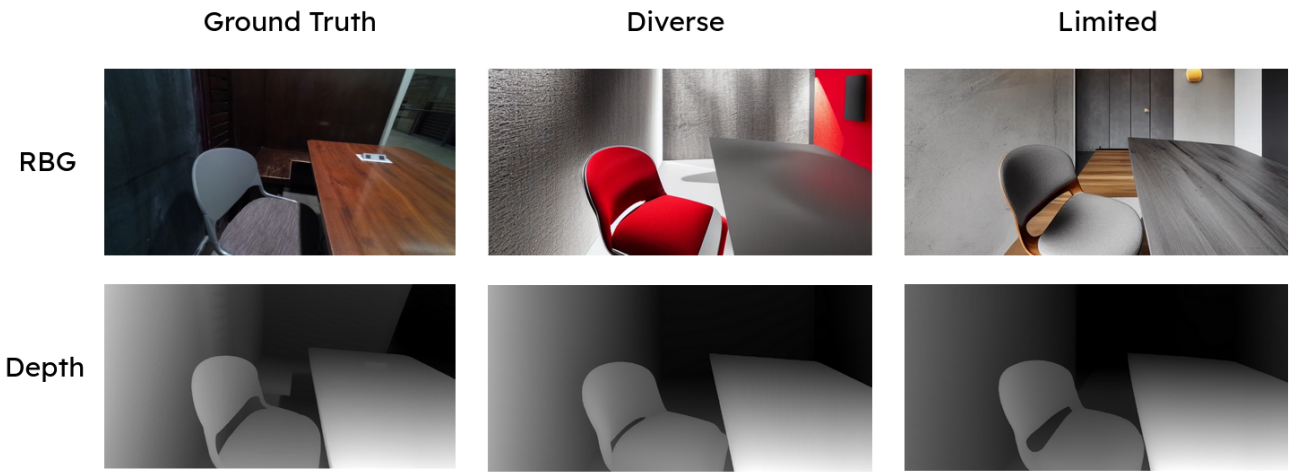

Figure 9. Good generations with $w = 0.8$ (including their Depth Anything depth map predictions)

our own dataset, we collect RGB and depth map pairs using an Azure Kinect DK [19]. We focus on two key principles we must follow for a fair comparison:

1. The datasets are of the same quality (particularly the depth maps)
2. Metrics should focus on conditional generation quality while also characterizing samples' diversity and adherence to depth maps

For our experiments, we chose to fine-tune a Stable Diffusion 1.5 backbone on both datasets using a ControlNet setup.

4.2. Experimental Setup

Hardware setup. The device we used to capture image and depth map pairs is an Azure Kinect DK, which uses a 12MP CMOS sensor with rolling shutter for RGB captures and a 1MP Time-of-Flight (ToF) sensor for depth. On top of depth, the device also has the ability to capture audio, accelerometer and gyroscopic data, but we only include experimental results for conditioning exclusively on depth. To interface with the Kinect, we use a Canakit Raspberry Pi 4 running the latest version of Ubuntu. Since the Pi uses an ARM instruction set and not x86, we compiled the Kinect libraries from scratch, installing necessary drivers along the way. To capture single recordings, we wrote a Python script to save one-second recordings in a Microsoft-defined mkv format using k4arecorder then extracted the first non-corrupt frame pair to two png files using PyK4a.

Data collection. To collect data from a single environment, we chose to scan sections of the Bahen Centre for Information Technology at the University of Toronto St. George campus since it has several floors with differing styles, but it does not seem nearly as diverse as the images in SUN RGB-D. Since SUN RGB-D contains approximately ten thousand images, we collected the same number in Bahen in approximately 4 hours of scanning. For both datasets, we leave prompts blank and condition on depth alone. The raw SUN RGB-D (diverse) and Bahen (limited) datasets are depicted in Figures 5 and 7. All images and depth maps are later downsampled to 512×512 resolution.

Datasets comparison. Since hardware is not consistent across our two datasets, we need to ensure the two sets of depth maps are of the same quality. Table 1 shows this comparison where completeness is the percentage of pixels that are not nan (sensor read a value), depth range is average depth range per map after globally normalizing (across the whole dataset) to $[0, 1]$, entropy is the entropy of both dataset's distributions approximated as 256 bin histograms, and noise level is the average standard deviation over all 5×5 patches in each image (with stride 5).

To ensure similar quality, we force completeness to 100, tune the other metrics to be equal, and sanity check the results qualitatively. For SUN RGB-D, this involved a simple thresholding procedure: normalizing all depth maps to $[0, 1]$, interpolating all values larger than 0.8, then unnormalizing back. The results of this processing are shown in Figure 6. For our Bahen dataset, this process was more involved: we used five consecutive iterations of non-local means with a gradient threshold of 10, outlier threshold of 15, filter strength of 15, template window size of 5, and search window size of 25. Then, we interpolated outliers (difference threshold of 100) in each 3×3 area to eliminate remaining fragmented salt and pepper noise. The results of this process are shown in Figure 8.

4.3. Training Setup

Both models were trained using a batch size of 4, one gradient accumulation step, a learning rate of 1×10^{-5},

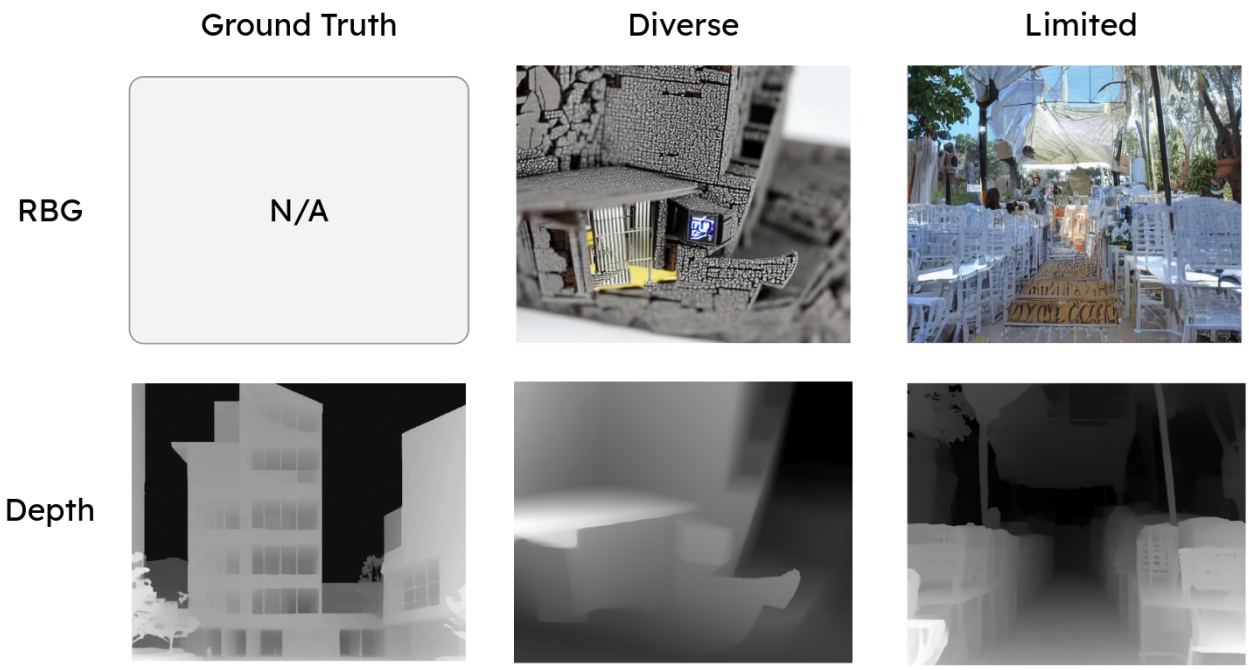

Figure 10. Bad generations with $w = 0.65$ (including their Depth Anything depth map predictions)

Metric	Diverse	Limited
FID	11.34	9.782
LPIPS	0.1400	0.1536
Depth Adherence	45.67	61.28

Table 2. Comparison of depth map quality metrics between raw SUN RGB-D and our raw Bahen dataset

and mixed fp16 precision. We trained both models for $\sim 250,000$ gradient descent steps on a single RTX6000 each, which took over a day.

4.4. Results and Discussion

Depth adherence. To compute adherence to depth maps, we decided to feed final samples through Depth Anything [37], an image to depth map foundational model, then calculate the squared error between the output and the original depth map we conditioned on. To test this, we create a third depth map and image pairs dataset composed of 1000 ImageNet [7] images along with their Depth Anything predictions.

Generation quality. Instead of only computing FID scores [11] on a test set for each dataset, which is common for unconditional models, we also report learned perceptual image patch similarity (LPIPS) [39] to measure out-of-distribution generation quality using the same ImageNet dataset as our depth adherence metric.

Final results. The scores we end up with are shown in Table 2 where sampling is performed using $w = 1$ with (11). The diverse model trained on SUN RGB-D beats out our limited model trained on the Bahen dataset in terms of LPIPS, but surprisingly comes up short in terms of FID. Figures 9 and 10 show cherry-picked samples from both models. Figure 9 shows the highest quality generations we could find where the ground truth is taken from an unseen scene in Bahen. Figure 10 shows one of the worst generations we could find where neither model came even close to handling one particular out-of-distribution depth map.

4.5. Limitations

Our experiments had several clear limitations such as

1. Dataset size: typical depth ControlNet dataset sizes use at least 100,000 images, ten times the number of ours
2. Only using two datasets instead of a breadth of datasets ranging from extremely limited to very diverse
3. The difficulty of quantifying depth diversity
4. Lack of experiments conducted on different ControlNet backbones: we only use the Stable Diffusion 1.5 U-Net

Lastly, if we were to redo our experiments, we would consider revamping our hardware setup, such as using a ToF sensor with a larger range or a gyroscope to stabilize handheld RGB captures.

5. Conclusion

In this work, we explored the feasibility of conditioning diffusion-based generative models on monocular depth maps without relying on extensive data diversity. Using ControlNet atop a Stable Diffusion 1.5 backbone, we empirically compared generation quality between models fine-tuned on a highly diverse dataset (SUN RGB-D) and a constrained, scene-limited dataset collected from Bahen. Despite the limited scene variety, the Bahen-conditioned model exhibited comparable—and in some metrics, superior—performance, particularly in depth adherence and FID. These findings support our initial hypothesis: depth maps, even when sourced from a restricted setting, can provide a sufficiently expressive conditioning signal for diffusion models to generate coherent and diverse imagery using only a pre-trained unconditional model. This suggests a promising path for efficient conditional generation using low-cost, domain-specific data, with potential for broader application in contexts where large-scale data collection is impractical. We are particularly excited by the implications this direction holds for the feasibility of conditional video generation models.

References

[1] Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin. Spatext: Spatio-textual representation for controllable image generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 18370–18380. IEEE, 2023. 2

[2] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation, 2023. 2

[3] Shane Barratt and Rishi Sharma. A note on the inception score, 2018. 3

[4] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 60–65 vol. 2, 2005. 4

[5] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986. 1

[6] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation, 2018. 2

[7] 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. 6

[8] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. 2, 3

[9] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors, 2022. 2

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 2

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. 3, 6

[12] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. 2, 3

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020. 1

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 2, 3

[15] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *J. Mach. Learn. Res.*, 6:695–709, 2005. 1

[16] Adam Janoch, Sergey Karayev, Yangqing Jia, Jonathan T Barron, Mario Fritz, Kate Saenko, and Trevor Darrell. A category-level 3-d object dataset: Putting the kinect to work. In *ICCV Workshop on Consumer Depth Cameras for Computer Vision*, 2011. 4

[17] Minguk Kang and Jaesik Park. Contragan: Contrastive learning for conditional image generation, 2021. 2

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. 2

[19] Microsoft Corporation. Azure kinect dk. https://azure.microsoft.com/en-us/products/kinect-dk/, 2019. Available at https://azure.microsoft.com/en-us/products/kinect-dk/. 5

[20] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W. Battaglia. Generating images with sparse representations, 2021. 3

[21] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021. 3

[22] 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, 2022. 2

[23] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 2

[24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 3, 4

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 4

[26] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation, 2023. 2

[27] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2016. 3

[28] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision (ECCV)*, pages 746–760. Springer, 2012. 4

[29] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. 1

[30] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. 3

[31] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020. 1

[32] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *CoRR*, abs/2011.13456, 2020. 1

[33] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders, 2016. 2

[34] Andrey Voynov, Kfir Aberman, and Daniel Cohen-Or. Sketch-guided text-to-image diffusion models, 2022. 2

[35] Jiajun Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1625–1632. IEEE, 2013. 4

[36] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes, 2016. 2

[37] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data, 2024. 6

[38] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 1

[39] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. 6

[40] Bolei Zhou, Agata Lapedriza, Jiajun Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014. 4