Sparse to Dense Depth Completion using a Generative Adversarial Network with Intelligent Sampling Strategies

Md Fahim Faysal Khan
The Pennsylvania State University
University Park, PA, USA
mzk591@psu.edu

Nelson Daniel Troncoso Aldas
The Pennsylvania State University
University Park, PA, USA
ndt5054@psu.edu

Abhishek Kumar
The Pennsylvania State University
University Park, PA, USA
azk6085@psu.edu

Siddharth Advani
Samsung Electronics America
Plano, TX, USA
s.advani@samsung.com

Vijaykrishnan Narayanan
The Pennsylvania State University
University Park, PA, USA
vijaykrishnan.narayanan@psu.edu

ABSTRACT

Predicting dense depth accurately is essential for 3D scene understanding applications such as autonomous driving and robotics. However, the depth obtained from commercially available LiDAR and Time-of-Flight sensors is very sparse. With RGB color guidance, modern convolutional neural network (CNN) based approaches can recover the missing depth information. However, there could be scenarios such as low-light environments where it might be difficult to get an associated RGB image with the sparse depth. In this work, we propose a Generative Adversarial Network (GAN) that can accurately predict the dense depth using only sparse samples without any RGB inputs. Generally, the sparsity in the depth samples is uniformly distributed and cannot guarantee capturing all intricate details. In this study, we also explore different variants of sparse sampling strategies from uniform to feature based directed sampling. We find that feature based intelligent sampling enjoys better compression ratio without sacrificing intricate details, saving data communication bandwidth. Compared to uniform sampling, depending on how aggressively the directed sampling is done, we observe about 3% to 25% reduction in size. We can easily reduce the size by 8% with directed sampling without sacrificing the reconstruction accuracy. Although such directed sampling strategies are not readily available with commercially viable depth sensors, we believe that our study paves the way for future intelligent sensing and sampling strategies. To further investigate data reduction and reconstruction accuracy trade-offs we deploy our GAN to generate higher resolution dense depth from 4× smaller sparse samples. With slight decrease in accuracy, our GAN is able to recover the depth successfully which shows great promise in edge Internet of Things (IoT) applications where we have very tight constraint on data transmission bandwidth. Our source code along with examples is available at: https://github.com/kocchop/depth-completion-gan

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475688

CCS CONCEPTS

• **Machine learning** → *Generative Adversarial Networks (GAN)*; • **Computing methodologies** → *Image compression*; *Computer Vision*; • **Depth Sensors** → *Directed LiDAR*.

KEYWORDS

Depth Completion; GAN; Image Compression; Sensor Sampling

ACM Reference Format:
Md Fahim Faysal Khan, Nelson Daniel Troncoso Aldas, Abhishek Kumar, Siddharth Advani, and Vijaykrishnan Narayanan. 2021. Sparse to Dense Depth Completion using a Generative Adversarial Network with Intelligent Sampling Strategies. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475688

1 INTRODUCTION

Sensing range accurately is critical for real-time mapping and surveillance tasks such as simultaneous localization and mapping (SLAM), autonomous and semi-autonomous guidance for vehicles, path planning for space crafts and similar robotic applications. Accurate and dense depth measurements is also extremely useful for object avoidance, detection, 3D modeling and reconstruction tasks. Mainstream range detection sensors include LiDAR scanners, *time of flight* (ToF) sensors and RGBD cameras. Amongst these, LiDAR sensors provide accurate depth information for longest range and hence are the most convenient for outdoor environments. However, these sensors suffer from some inherent limitations such as low frequency and sparse samples. Standard LiDAR sensors operate at 5-15 Hz and only provide 32 or 64 scanlines across the whole image [9]. This type of sampling results in heavily sparse depth images whereas most of the applications require dense and accurate depth information. Hence, predicting dense and accurate range is an active area of research.

Generative Adversarial Networks (GAN) are seen to perform exceptionally well in single image super resolution i.e. constructing accurate and detailed images from low resolution images [16, 25]. With enough training data, GANs can learn to generate samples adhering to the parent distribution. Consequently, they have been a popular choice for tackling problems requiring generating images. Convolutional neural networks (CNNs) on the other hand can produce state-of-the-art sparse to dense depth completion results but

they require additional RGB data [19, 22]. In this work, we aim to produce accurate dense depth images only from sparse examples without any RGB information. Consequently, in this study, we leverage a GAN to predict dense depth without any RGB guidance or prior and show that they can produce competitive results compared to the state-of-the-art CNNs with a smaller memory footprint and lower latency. We also deploy a soft visual attention mechanism and introduce normal loss as perceptual loss indicated in [25].

While high resolution and more detailed data generally gives better accuracy, it also pushes the limits of communication bandwidth. With better connectivity and recent advancement in Internet of Things (IoT), we are generating more data than ever. More than 300 million photos get uploaded to Facebook everyday and on average, 300 hours of video are uploaded on YouTube every minute [7]. Storage and transmission of these multimedia contents requires huge amount disk space and network bandwidth. Therefore, data compression techniques which can save storage memory and transmission bandwidth without affecting the quality are becoming more and more significant than ever. The basics of image compression lies in reducing data redundancy and saving only what is needed. A uniformly sampled image contains most randomness thus providing less room for compression. On the contrary, a directed sampling strategy can readily help data compression by reducing the randomness. Moreover, having more data in complicated region contributes towards higher reconstruction accuracy. In this work we show how directed depth samples can help improving the compression ratio without impacting the quality. To summarize, in this paper, our key contributions are as follows:

(1) We propose a Generative Adversarial Network (GAN) as a depth reconstruction framework which provides competitive predictions only from sparse depth samples compared to what state-of-art CNNs achieve using both RGB image and sparse depth samples. Our proposed architecture utilizes less storage both during training and inference.
(2) We explore different sampling strategies other than uniform sampling and show compression benefits. The sampling strategies mainly consist of feature based sampling where features are extracted using SURF [1].
(3) Our proposed GAN can also be used to generate super resolution depth images from much smaller sparse samples, further increasing storage benefits.

2 RELATED WORKS

2.1 Depth Completion

The depth completion technique aims at constructing the 3D image by filling the holes in the existing 3D frame. Traditional approaches like dedicated kernel development to construct the missing pieces of the depth image [6] are very scene-specific and are difficult to generalize. A relatively newer CNN-based approach tends to outperform classic approaches and also shows good generalization characteristics. Many state of the art deep neural networks like [10, 18, 19, 22, 23, 26] leverage both sparse depth data and RGB images to predict depths. Sparse-to-Dense [19] suggested an encoder-decoder architecture for dense depth prediction. Later, they extended it by proposing a self-supervised depth completion mechanism that does not require ground truth labels [18]. FusionNet [23] uses two separate branches for global and local feature extraction with RGB guidance in order to predict depth. The two branches also predict separate confidence maps that are used in fusing two predictions in order to generate the final depth. In this study, we refer to FusionNet as "Sparse Depth". Hu et al. [10], highlights the importance of the balanced fusion of the color and depth to gain the performance boost in dense depth map generation task. It also introduces the idea of a geometric convolution layer that performs a fusion of different modalities leading to better accuracy. Xiong et al. [26] proposed a graph convolution model which tries to exploit the neighborhood relationship of 3D points by the usage of dynamic construction of local neighborhood instead of traditional square kernels. One key aspect worth mentioning is that most of these CNN based approaches require multiple training stages. For example, DeepLiDAR [22] first trains a network to generate surface normals from sparse depth data. Another pipeline is separately trained to produce depth images only from RGB data. Finally, these two pipelines are fused and trained end to end to generate the final depth image. On the contrary, our proposed GAN is essentially a single architecture trained in an end to end fashion from the very beginning. More importantly, while other approaches use RGB guidance for the depth completion task, our model uses depth images alone to accomplish the same.

2.2 Sampling Strategy

Depth maps can be generated either by directly measuring distance with sensors such as laser scanners (LiDAR), time-of-flight cameras and ultra wide band radars, or from inferring distance from RGB images. There are two main approaches for inferring distance from images. The classical approach is using parallax, i.e. displacement of objects between images. Modern approaches are able to infer distance from a single image using deep learning algorithms [10, 19]. In contrast, depth sensing devices can directly obtain depth measurements with sensor readings; the advantage of these technologies is the accuracy they provide when compared to RGB image systems. Junming et al. [27] proposed a cost effective solution by fusing the image coming from LiDAR and stereo matching algorithm to produce high quality depth map.
Recently the application of 3D imaging systems has increased significantly because of the advancement it gives in many day-to-day jobs. In order to balance between speed, power consumption, and image resolution, imaging hardware tends to work in sparse depth and uses the depth completion technique to convert from sparse to dense. Different sampling strategies have been explored to reduce the number of depth sample points. Most relevant to our work is Alexander et al. [2] who used Poisson-disc [3] sampling technique by adding a learnt four layer visual attention network. In contrast to their approach, which is consistent with furthest point sampling approach, our technique densely samples point around features extracted by a light weight feature extractor. Hence, our approach is beneficial for compressibility of the sampled data which is the focus of our evaluation. Prior work has not investigated compression.

2.3 Attention Mechanism

Convolution on higher resolution image is a compute expensive operation and lately has received a lot of attention in deep learning literature. In order to expedite the convergence, many recent works [8, 17] used attention map to direct the focus of the network

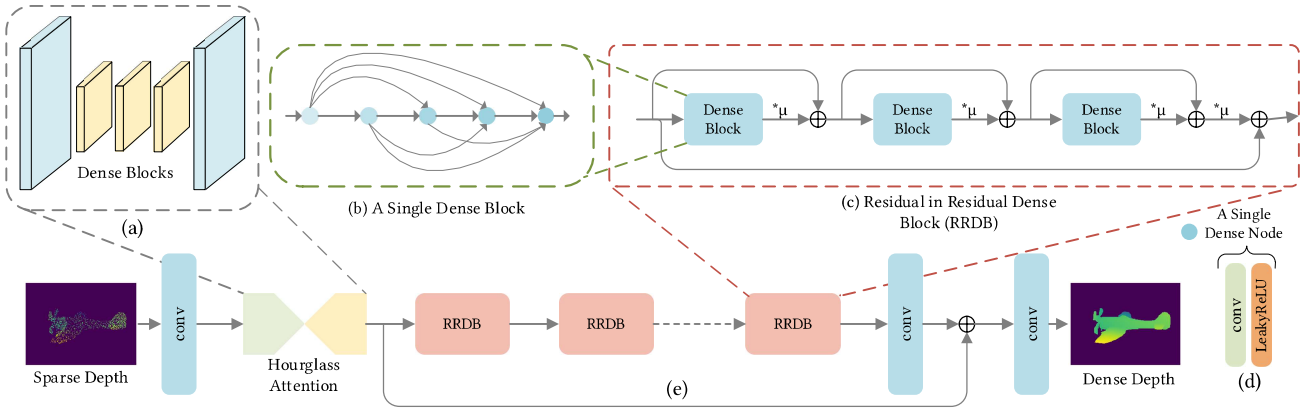

Figure 1: Our overall GAN Architecture. (a) shows the structure of the hourglass attention module, (b) shows a Dense Residual Block configuration consisting of convolution layers followed by LeakyReLU as showed in (d) except for the last one. (c) illustrates the structure of a RRDB block consisting of 3 Dense blocks. Finally, (e) gives the overview of the whole architecture.

towards certain regions and increase the performance of convolution operator. The most common structure of an attention module is an encoder-decoder structure. The encoder extracts richer context from the input and the decoder eventually learns where to pay attention. In addition to that, attention mechanisms are seen to improve the prediction accuracy. In this study, we also introduce a soft attention module for similar purposes.

3 METHOD

In this section, we briefly describe the problem statement leading towards the GAN architecture. We also describe the most popular image compression algorithms and investigate how randomness in data affects their compression ratio.

3.1 Problem Statement

The basic setting of a GAN consists of two networks naming the **generator** and the **discriminator**. The generator learns to create samples identical to an input distribution and the discriminator learns to distinguish between real and generated fake data. Eventually, the training reaches convergence when the generator becomes successful in fooling the discriminator. In our problem setting, the input data distribution is the sparse depth image, I^{SP} and we aim to predict the dense depth, I^{DN} from it. The ultimate goal is to train a generator function, G which can produce dense depth samples from the sparse inputs. Generally, the generator, G is a multi-layer perceptron network, in our case a CNN G_{θ_G} having the parameters θ_G. We assume the generator loss function is defined as L^G. Then for training samples, $I_n^{SP}, n = 1, ..., N$ with corresponding $I_n^{DN}, n = 1, ..., N$, we aim to solve:

$$\hat{\theta}_G = \arg\min_{\theta_G} \frac{1}{N} \sum_{n=1}^{N} L^G \left(G_{\theta_G} \left(I_n^{SP} \right), I_n^{DN} \right) \tag{1}$$

It is to be noted that the loss function L^G is a weighted combination of different loss components which include the discriminator loss. The other loss components focus on definitive characteristics of the desired dense depth image. The details regarding the loss function is explained in detail below.

3.1.1 Adversarial Loss. The discriminator, $D_{\theta_D}^{RA}$ is also a CNN based architecture with parameters θ_D. Following [25], we use a Relativistic average Discriminator (RaD) explained in [14], denoted as D^{RA}. The sole purpose of the discriminator is to identify generated fake samples and the generator tries to fool it making it a two player min-max game. While the standard discriminator estimates the probability of its input being real or fake, the relativistic discriminator operates on two images, one is the generated one $\hat{I}^{DN} = G_{\theta_G}(I^{SP})$ and another is real I^{DN}. It tries to predict the probability of the real one, I^{DN} being relatively more realistic than the generated $\hat{I}^{DN}$. This leads us to the discriminator loss defined as:

$$\begin{aligned} l_{Ra}^{D} = -\mathbb{E}_{I^{DN}} \left[\log \left(D^{Ra} \left(I^{DN}, \hat{I}^{DN} \right) \right) \right] - \\ \mathbb{E}_{\hat{I}^{DN}} \left[\log \left(1 - D^{Ra} \left(\hat{I}^{DN}, I^{DN} \right) \right) \right] \end{aligned} \tag{2}$$

Similarly, we get the adversarial loss for the generator in a symmetrical form:

$$\begin{aligned} l_{Ra}^{G} = -\mathbb{E}_{I^{DN}} \left[\log \left(1 - D^{Ra} \left(I^{DN}, \hat{I}^{DN} \right) \right) \right] - \\ \mathbb{E}_{\hat{I}^{DN}} \left[\log \left(D^{Ra} \left(\hat{I}^{DN}, I^{DN} \right) \right) \right] \end{aligned} \tag{3}$$

where, $\mathbb{E}_{I^{DN}}[\cdot]$ denotes the average taken out for all the real samples inside the mini-batch.

3.1.2 Normal Loss. In addition to the adversarial loss, we also use normal loss as the perceptual loss for the generator network. In ESRGAN [25], the authors used a pretrained VGG backbone to calculate a perceptual loss on the generated data. This approach is not suitable in our case since we are not dealing with RGB data. On the other hand, the calculation of image normals from a depth map is quite straightforward making it the most meaningful intermediate depth representation. Hence, we leverage the mismatch between generated and real depth image normals and use that as

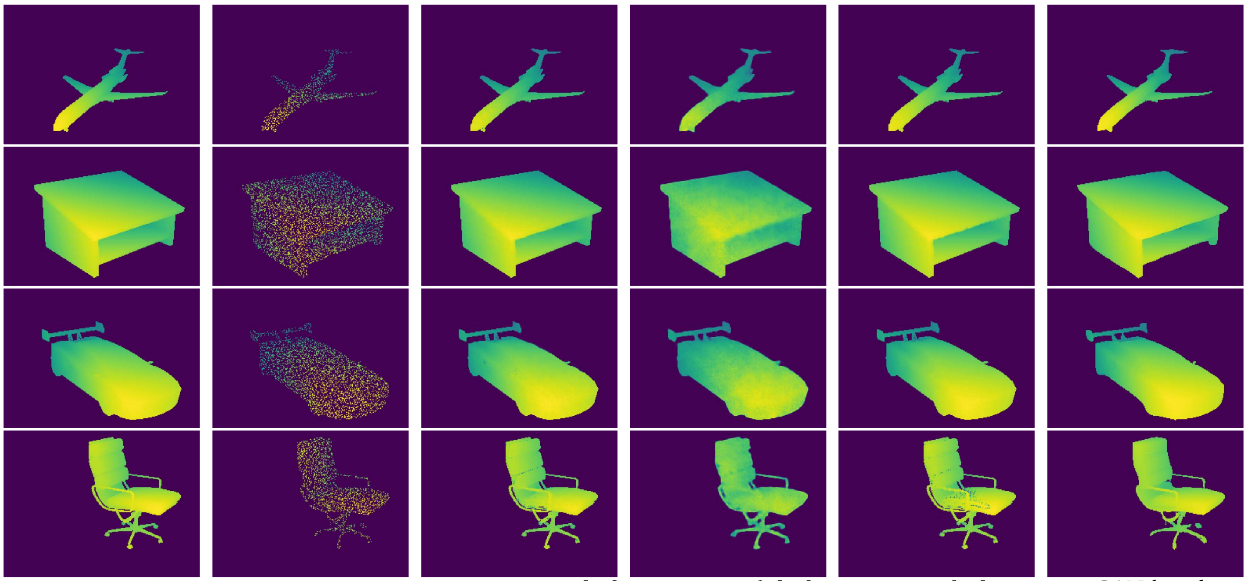

Figure 2: Qualitative comparison among the state-of-the-art depth completion architectures. The sparse input has been uniformly sampled in this case. Our GAN has visibly produced much better results compared to others.

the perceptual component inside generator loss. If $\nabla^{DN}, \hat{\nabla}^{DN}$ are the corresponding gradient vectors of ground truth I^{DN} and the prediction $\hat{I}^{DN}$, then the normal loss over $n = 1, ..., N$ samples can be presented as:

$$l_{\text{normal}} = \frac{1}{n}\sum_{i=1}^{n}\left(1 - \frac{\langle \nabla_i^{DN}, \hat{\nabla}_i^{DN} \rangle}{\|\nabla_i^{DN}\| \left\|\hat{\nabla}_i^{DN}\right\|}\right) \tag{4}$$

where, $\langle \cdot \rangle$ denotes the dot product of the gradient vectors and $\|\cdot\|$ denotes the norm of the corresponding vectors. For depth images, normal loss is exceptionally helpful to capture finer and intricate details. That is why, we use it as a perceptual loss for the generator.

3.1.3 Pixel Loss. It is the most straightforward pixel to pixel loss, also known as l_1 loss and defined as:

$$l_1 = \mathbb{E}_{I^{SP}} ||G_{\theta_G}(I^{SP}) - I^{DN}||_1 \tag{5}$$

Finally, we get the total generator loss as:

$$L^G = l_{normal} + \alpha * l_{Ra}^G + \beta * l_1 \tag{6}$$

The α and β are the scaling factors in order to balance different loss components. By tweaking these two hyperparameters we can particular favor certain kind of features while training the network. However, in general training settings, these two are kept at constant values.

3.2 Network Architecture

In this section, we briefly describe the basic components of the proposed network architecture.

3.2.1 Dense Residual Blocks: The core building block of our GAN architecture is a densely connected residual block [11]. Dense networks can effectively differentiate between already preserved information in the network and the newly added information improving the overall collective knowledge. For larger networks with many layers, the dense layer eases the information and gradient flow.

3.2.2 Hourglass Attention Mechanism: The hourglass network enables capturing attributes of image at different scale allowing the model to gain more insight on the context by increasing the number of features used for training as compare to the other networks. The hourglass is constructed by performing convolution and max pool operation in each layer along with down sampling of images so that extracting features is less compute expensive. After reaching the lowest granularity, the network begins to upsample the images and consolidates all the features extracted so far [21]. The difference in our implementation is that we use the dense residual blocks as opposed to normal convolutional blocks inside the hourglass structure.

3.2.3 Residual in Residual Dense Blocks (RRDB):. Residual in Residual Dense Blocks (RRDB) [25] helps boost the performance by increasing the number of connection and is proven to perform well in reconstructing finer details. In our study, we aim to generate dense depth from non-uniform sparse samples which is why we use the RRDB as the basic block of our generator architecture.

3.3 Image Compression

Depth images or depth maps express distance information about the pixels in the image with respect to the viewpoint of the image taken. Nowadays, depth images are used in a wide array of applications

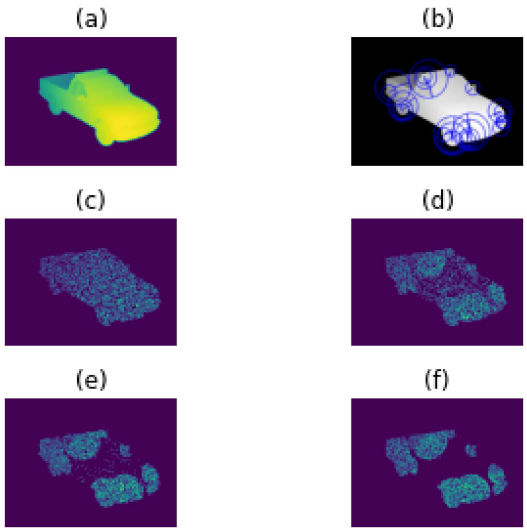

Figure 3: (a) Original image, (b) image with SURF features highlighted, and (c)-(f) sampled images with varying weights = 1, 2, 4, and 8 respectively.

from medical imaging, augmented reality, photography to robotics. In augmented reality, for example, depth information is leveraged to create more immersive experiences because it enables occlusion and collision of virtual objects. In robotics, depth imaging is used for navigation and localization.

Due to storage or bandwidth restrictions, such applications might need to reduce the size of data they receive, store and generate. In order to satisfy these constraints without compromising the effectiveness, these applications employ compression algorithms. Image compression algorithms are concerned with the minimization of bits needed to represent an image [12], as to minimize bandwidth and storage usage. In general, compression techniques like the ones used in image formats such as PNG, TIFF, and JPEG take advantage of redundancy, irrelevancy, predictability and the statistical distribution of the data [13]. Take PNG for example. When converting to this format, encoders leverage DEFLATE compression [24], which reduces redundancy of data by searching duplicate symbols and replacing them with pointers. And finally, the symbols get replaced by new ones depending on their statistical frequency [5, 28]. Also, as part of the conversion process, PNG encoders apply some kind [24] of filtering before compression so as to make the data more compressible. In particular, delta filtering reduces the number of symbols in an image by expressing them as the difference of neighboring pixels. This makes an image more compressible since a scheme like DEFLATE will eliminate duplicate data and it will decrease the number of symbols needed to represent an image.

An important characteristic of data that is highly compressible is that it should have a lower entropy. In other words, its predictability and regularity should be high. The more random a sample of data is, the less compressible it is. In this work we exploit this concept by demonstrating that if instead of uniformly, i.e. randomly sampling data, we direct our sampling to certain areas of the image, then we can reduce the size of the image without sacrificing reconstruction accuracy.

Table 1: Comparison of reconstruction accuracy

Architecture	RMSE	MAE	iRMSE	iMAE
Sparse-to-Dense [19]	0.02	0.01	0.42	0.03
Sparse Depth [23]	0.12	0.03	36.24	0.15
PENet [10]	0.02	0.004	1.99	0.03
GAN-based	0.17	0.016	0.86	0.01

Table 2: Model size, data volume and latency Comparison

Architecture	Model Size (MB)	Data Volume Train/Infer	Inference Time (fps)
Sparse-to-Dense [19]	299	1×/1×	40
Sparse Depth [23]	30	1×/1×	7.14
PENet [10]	504	1×/1×	26.32
GAN-based	116.5	0.6 × /0.4×	17.24

4 EXPERIMENTAL SETUP

Dataset Generation: For our experiments, we choose 2D rendered images of a subset of models from the ShapeNet dataset [4]. Our training set comprises of 128K randomly chosen samples and the validation set contains 1.2K samples which are distinct from the training ones. Each of these samples has their associated RGB image and 16-bit dense depth maps. In order to get sparse samples, we sub-sample the dense depth image. While sampling the dense depth maps, we leverage several sampling schema described in Section 4.1.

Training Details: Following a similar training strategy as [25], we train the generator model with only the pixel or $l1$ loss for about 250 iterations. After that, we start incorporating other two loss functions as well. The first initialization step helps to avoid local optima and also it prevents from sending extreme false examples to the discriminator. For the generator, we deploy 17 RRDB blocks for the depth completion task. In the depth super-resolution experiment, 23 RRDB blocks are used.

As an optimizer, we use Adam [15] for both the generator and discriminator model with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. As for the loss co-efficients, we set the adversarial loss factor, $\alpha = 5e-3$ and the pixel loss factor, $\beta = 1e-2$. The total training framework is implemented in PyTorch. For training, 4 Tesla V100 GPUs are used where it takes roughly four days to finish 11 epochs.

4.1 Sampling Strategies

Here we explore the regime of non-uniform sampling. As mentioned in Section 3.3, non-uniform sampling reduces the randomness resulting in lower entropy. Hence, non-uniform sampling helps improve the data compression. Moreover, uniform sampling does not always guarantee capturing all the details. On the other hand, if we could do some sort of non-uniform sampling allocating more sampling for important parts of the scene, we supposedly should have a better detailed view. In order to find out the interest points

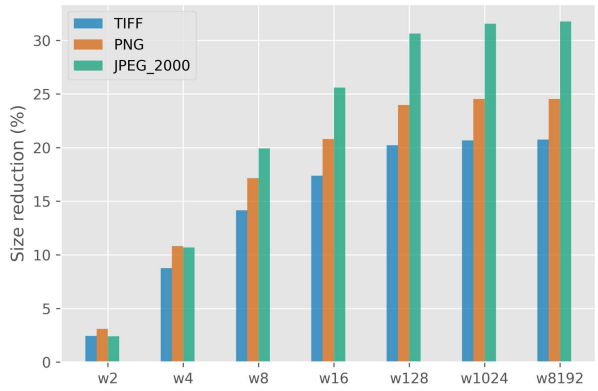

Figure 4: Size reduction in % for $w = 1, 2, 4, 8, 16, 128, 1024, 8192$ for images in TIFF, PNG and JPEG-2000.

inside an image, we leverage classic feature extraction techniques like SURF [1]. Depending on detected features, we deploy several non-uniform sampling strategies such as:

(1) **Uniform Sampling:** Here we uniformly sample the images targeting for a fixed percentage of the valid pixels in each image. We assign a weight mask for every valid pixel and for uniform, the weight is set to w=1.
(2) **Weighted Sampling:** Next we assign different weights for different pixels. The points inside SURF detected feature are associated with a higher weight value compared to other points. Finally, weighted sampling is performed to select the desired number of pixels. For weighted sampling, we experimented with w=2,4,8 etc. A weighted point inside a SURF detected feature with w=2 means that the point has twice the chance of being selected compared to others.
(3) **Hybrid sampling:** In case of hybrid sampling, first we select a small portion of the points uniformly and then the remaining portion is sampled by weighted sampling as mentioned above.

An important point to note here is that for all these sampling strategies, the point volume is kept constant.

5 RESULTS

In this section, we discuss the results. We perform both qualitative and quantitative comparison of our proposed GAN with contemporary depth completion works. For quantitative comparison, we choose the following evaluation metrics: 1) root mean square error (RMSE), 2) mean absolute error (MAE), 3) root mean square of the inverse depth (iRMSE) and 4) mean absolute error of the inverse depth (iMAE). Since no prior works used the sub-sampled sparse images from ShapeNet, we train the baselines ourselves on the rendered ShapeNet dataset and report the best results.

5.1 Depth Reconstruction Accuracy

In this experiment, we compare predictions from our GAN with other state-of-the-art methods. For this study, we choose the validation set with uniformly sampled images. Fig. 2 shows the qualitative results for the methods discussed. Visibly, our method generates sharper images very close to the ground truth compared to others. Table 1 shows the performance of our method compared to other state-of-the-art depth completion techniques quantitatively. Although, our network's prediction accuracy is not the best, it produces competitive results. The reason for the decrease in performance is because our GAN generates the outputs from the sparse depth only without any input from the RGB.

Table 3: Reconstruction accuracy for different sampling strategies across different frameworks

Sampling Strategy	Sparse to Dense [19]	Sparse Depth [23]	PENet [10]	GAN based	Size (%) Reduction
w=1	0.02	0.12	0.02	0.17	0.00
w=2	0.02	0.12	0.02	0.18	3.39
w=4	0.02	0.12	0.02	0.18	5.30
w=8	0.02	0.12	0.03	0.23	10.78
hybrid	0.02	0.12	0.02	0.18	6.55

5.2 Training and Inference Storage Benefits

Our GAN generator model occupies less memory footprint compared to most other state-of-the-art models discussed here as showed in Table 2. More importantly, for training and testing it uses 40% and 60% less data respectively. This reduction in data volume has been possible as our GAN does not use RGB data. In terms of inference latency, our generator model has competitive throughput compared to others. By leveraging quantization and pruning based model compression techniques, we can further reduce the model size and inference complexity.

5.3 Compression Ratio for Directed Sampling

To validate the compression benefits of weighted sampling, we sampled a set of 128 depth images. For every image we included 25% of the points that belonged to the object in the image, i.e. not including the background. To decide which points to pick from this 25%, we crafted a function that assigned to every pixel a weight, w. The weight reflected the probability of it being taken, a higher weight means a higher probability of being taken. To choose which pixels would have a varying weight, we used a feature extraction algorithm; in our experiment we used SURF. We followed this procedure for $w = 1, 2, 4, 8, 16, 128, 1024, 8192$ and saved the images in PNG, TIFF and JPEG-2000. Figure 3 shows a sample object with high resolution, with features highlighted and sampled with different weights.

Our experimental results show that as the weight increased, the compression ratio of the images increased as well. Figure 4 shows the results for the three compression formats. In PNG for example, images with $w = 1$ had a mean size of 9.08 KB while an image with $w = 8192$ had a mean size of 6.60 KB. This means that compared to uniform sampling, i.e. $w = 1$, this set of images had a decrease of 25% in size. In general, our experiments show that images that had directed sampling applied had a higher compression ratio than images with uniform sampling.

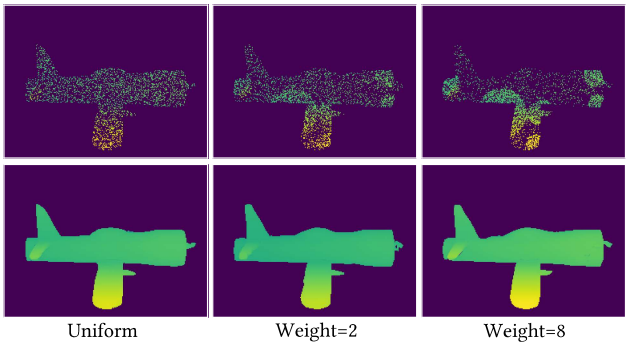

Figure 5: Reconstructed depth for directed sampled sparse inputs. With the increase of weights in directed sampling, we obtain better image compression ratio.

Table 4: Reconstruction accuracy for depth super resolution

Sampling Strategy	RMSE	MAE	iRMSE	iMAE	Size (%) Reduction
w=1	0.22	0.06	1342.67	0.90	90.60
w=2	0.28	0.04	6.15	0.04	90.68
w=4	0.31	0.06	2.58	0.05	90.87
hybrid	0.24	0.07	3731.81	1.4	90.82

5.4 Reconstruction Accuracy for different Non-uniform Sampling Strategies

We have discussed the compression benefits for directed sampling in the above as opposed to the uniform sampling. Here, we present the the effect of different sampling strategies on the prediction accuracy. In Table 3, we report the RMSE metric for different strategies mentioned in Section 4.1 across all the frameworks in consideration. Fig. 2 shows the reconstruction quality. Although the directed samples are generated with RGB guidance in this case, the RGB is not sent to the GAN, thus saving the transmission bandwidth. Table 3 shows a slight accuracy degradation for higher weights. However, with feature based directed sampling strategies, we obtain better compression ratio which is extremely useful for edge cases with limited network bandwidth.

5.5 Depth Super Resolution

We further investigate more intensive image compression schema where along with completing sparse depth maps, the spatial image resolution is also enhanced using our proposed GAN. In order to conduct this experiment, we first spatially downscale the image by 4× and then generate the sparse samples from that. We include two PixelShuffle upsampling layers at the end to generate the high resolution depth maps following similar approach as in [25]. Table 4 shows the reconstruction accuracy and compression benefits. The compression ratio is calculated with respect to high resolution sparse depth image. Such kind of aggressive compression could be useful for extremely stringent bandwidth requirements.

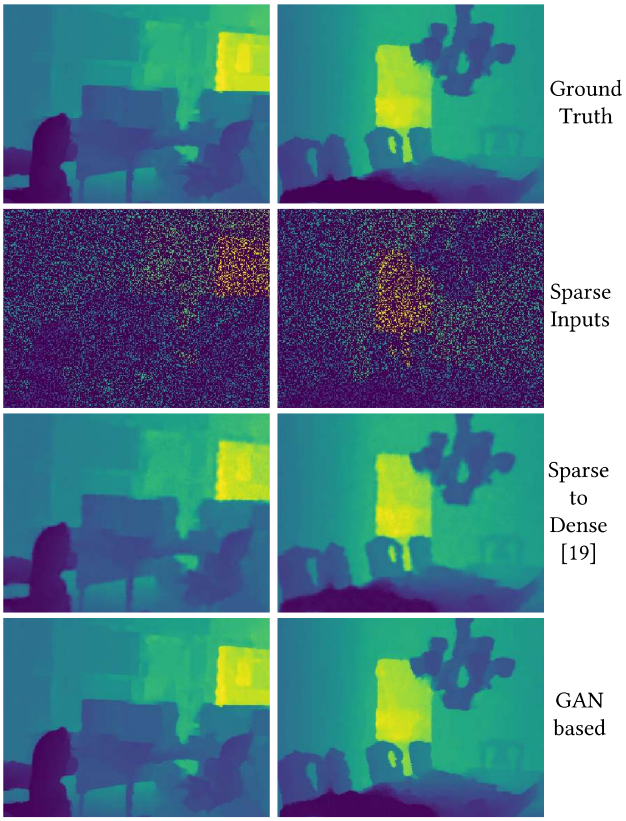

Figure 6: Qualitative study of reconstructed depth map on the NYU-depth-v2 [20] dataset. Our GAN-based approach generates sharper and less noisy image.

Table 5: Quantitative analysis on NYU-Depth-v2 dataset

Architecture	RMSE	MAE	iRMSE	iMAE
Sparse-to-Dense [19]	0.067	0.04	149.83	8.9
GAN-based	0.059	0.03	3505.91	12.02

5.6 Results on NYU-Depth-v2 Dateset:

For further validation, we evaluate our GAN-based approach on popular NYU-Depth-v2 [20] dataset. The NYU-Depth-v2 dataset consists of depth maps provided in real values with associated RGB images collected from 464 different indoor scenes. We use the official split as training (around 48K samples) and testing data (654 samples). The depth images were first down-sampled to half and then center-cropped to the size of 304x228 following [19]. Since, we use a separate sampling strategy, we cannot use the accuracy numbers directly from other studies. To alleviate the situation, we train Spare-to-Dense [19] from scratch on our version of sampled NYU-Depth-v2 dataset. We repeat the same for our GAN-based approach and measure the metrics. Table 5 shows the quantitative comparison on reconstruction accuracy. Although our approach suffers in iRMSE and iMAE metrics, it outperforms other work in RMSE and MAE metrics. It must be noted that we only use

Table 6: Effect of Hourglass Attention and Normal Loss

Generator	RMSE	MAE	iRMSE	iMAE
W/O HA	0.17	0.04	29.98	0.09
W/O NL	0.2	0.03	1.5	0.04
With HA + NL	0.17	0.016	0.86	0.01

HA = Hourglass Attention; NL = Normal Loss

sparse inputs whereas the other work uses RGB information as well. Qualitatively, the generated dense depth maps look comparatively sharper and less noisy. This experiment confirms the efficacy of our proposed GAN across other datasets as well.

6 ABLATION STUDY

In this section, we perform an ablation study of the proposed network architecture. We introduced an hourglass attention module in our generator architecture and also included the normal loss in the generator loss function. We would sequentially remove these components one by one and see what it does to the overall reconstruction accuracy both qualitatively and quantitatively in order to determine their efficacy.

6.1 Hourglass Attention Module

First, we remove the hourglass attention module from the generator pipeline keeping all other parts unchanged and train the network from scratch in a similar fashion as the baseline. The reconstruction accuracy of the network with and without the hourglass attention module is provided in Table 6. Clearly, the error increases, especially, in case of the inverse error metrics. While RMSE and MAE metrics provide a measure of how well the model is performing in farthest depth prediction, the inverse error metrics measure the accuracy of the nearest depth points. Unlike many other depth completion datasets, ShapeNet gives relative depth which is normalized for training and we use these normalized values in order to calculate all the metrics. Hence, a significant portion consists of fractions which is why the inverse error metrics play a significant role representing the overall performance. The hourglass attention module certainly improves the overall quality remarkably.

6.2 Normal Loss

One of our key loss functions while training the generator, is the normal loss given in 4 which serves the purpose of the perceptual loss mentioned in [25]. In order to prove its efficacy, we retrain the generator network from scratch removing the normal loss and only using the pixel loss and adversarial loss. The quantitative comparison on reconstruction accuracy after training with and without the normal loss is showed in Table 6. We observe a remarkable degradation in RMSE compared to others while the generator is trained without the normal loss. Fig. 7 shows the results qualitatively. We investigate the reason and observe that without the normal loss the edges of the objects are little distorted while using normal loss results in a smooth edge. Since the edges are mostly located at the farthest points, intuitively without the normal loss, it should degrade RMSE the most. While other methods use RGB guidance and are capable of inferring edge information from that, our GAN-based approach achieves the same by incorporating the normal loss during training.

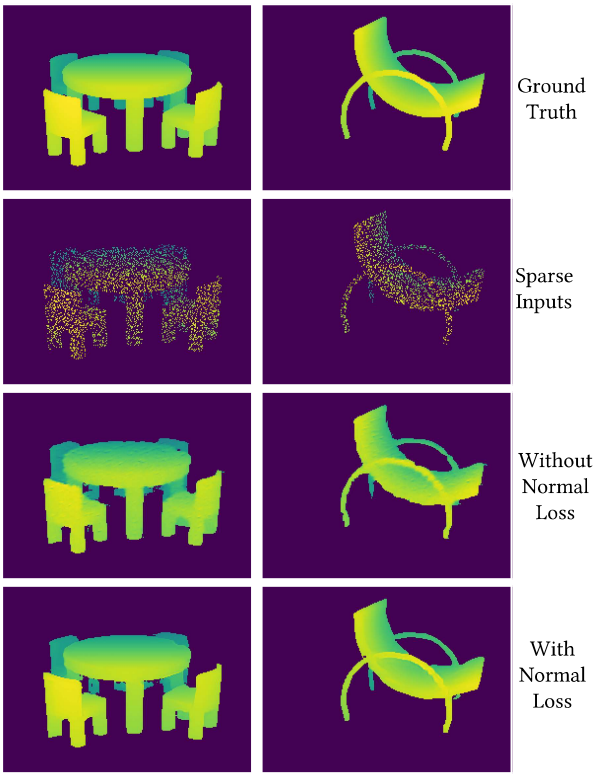

Figure 7: Qualitative comparison in between the reconstructed depth outputs from training with and without normal loss. Clearly, with normal loss, the model generates much better quality depth maps.

7 CONCLUSION

In this work, we presented a Generative Adversarial Network that can recover the dense depth from sparse samples without any RGB inputs. We also explore different variants of sparse sampling strategies from uniform to feature-based directed sampling. We find that feature-based intelligent sampling guarantees better detail providing better compression with less communication bandwidth requirement. Compared to uniform sampling, depending on how aggressively the directed sampling is done, we observe about 3% to 25% reduction in size. With directed sampling, we can reduce the size by 8% without sacrificing the reconstruction accuracy. In summary, our approach provides a holistic view of reducing the data volume from both the sampling and data transmission perspectives considering multiple metrics such as RMSE, training and inference data volume

ACKNOWLEDGMENTS

This work was supported in part by National Science Foundation (NSF) SOPHIA (CCF-1822923) and Center for Brain-inspired Computing (C-BRIC) & Center for Research in Intelligent Storage and Processing in Memory (CRISP), two of the six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

REFERENCES

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. 2006. SURF: Speeded Up Robust Features. In *Computer Vision – ECCV 2006*, Aleš Leonardis, Horst Bischof, and Axel Pinz (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 404–417.

[2] A. W. Bergman, D. B. Lindell, and G. Wetzstein. 2020. Deep Adaptive LiDAR: End-to-end Optimization of Sampling and Depth Completion at Low Sampling Rates. In *2020 IEEE International Conference on Computational Photography (ICCP)*. 1–11. https://doi.org/10.1109/ICCP48838.2020.9105252

[3] R. Bridson. 2007. Fast poisson disk sampling in arbitrary dimensions. *ACM Special Interest Group on Computer Graphics and Interactive Techniques Conference* (2007).

[4] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. 2015. *ShapeNet: An Information-Rich 3D Model Repository.* Technical Report arXiv:1512.03012 [cs.GR]. Stanford University — Princeton University — Toyota Technological Institute at Chicago.

[5] P. Deutsch. 1996. DEFLATE Compressed Data Format Specification version 1.3. https://doi.org/10.17487/RFC1951

[6] D. Doria and R. J. Radke. 2012. Filling large holes in LiDAR data by inpainting depth gradients. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 65–72. https://doi.org/10.1109/CVPRW.2012.6238916

[7] Dustin W. Stout. 2021. [online]. Available: https://dustinstout.com/social-media-statistics/. [Accessed: Apr-11-2021].

[8] Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. 2019. Attention Branch Network: Learning of Attention Mechanism for Visual Explanation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[9] Radu Horaud, Miles Hansard, Georgios Evangelidis, and Clément Ménier. 2016. An overview of depth cameras and range scanners based on time-of-flight technologies. *Machine vision and applications* 27, 7 (2016), 1005–1020.

[10] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. 2021. PENet: Towards Precise and Efficient Image Guided Depth Completion. arXiv:2103.00783 [cs.CV]

[11] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. 2017. Densely Connected Convolutional Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2261–2269. https://doi.org/10.1109/CVPR.2017.243

[12] A. K. Jain. 1981. Image data compression: A review. *Proc. IEEE* 69, 3 (1981), 349–389. https://doi.org/10.1109/PROC.1981.11971

[13] Uthayakumar Jayasankar, Vengattaraman Thirumal, and Dhavachelvan Ponnurangam. 2021. A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications. *Journal of King Saud University - Computer and Information Sciences* 33, 2 (2021), 119–140. https://doi.org/10.1016/j.jksuci.2018.05.006

[14] Alexia Jolicoeur-Martineau. 2019. The relativistic discriminator: a key element missing from standard GAN. In *International Conference on Learning Representations*. https://openreview.net/forum?id=S1erHoR5t7

[15] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. arXiv:arXiv:1412.6980

[16] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4681–4690.

[17] K. Li, Z. Wu, K. Peng, J. Ernst, and Y. Fu. 2018. Tell Me Where to Look: Guided Attention Inference Network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9215–9223. https://doi.org/10.1109/CVPR.2018.00960

[18] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. 2019. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 3288–3295.

[19] Fangchang Ma and Sertac Karaman. 2018. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 4796–4803.

[20] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. 2012. Indoor Segmentation and Support Inference from RGBD Images. In *ECCV*.

[21] Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked Hourglass Networks for Human Pose Estimation. In *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Springer International Publishing, Cham, 483–499.

[22] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. 2019. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3313–3322.

[23] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. 2019. Sparse and Noisy LiDAR Completion with RGB Guidance and Uncertainty. In *2019 16th International Conference on Machine Vision Applications (MVA)*. IEEE, 1–6.

[24] W3C. [n.d.]. https://www.w3.org/TR/2003/REC-PNG-20031110/#10Compression

[25] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. 2018. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 0–0.

[26] Xin Xiong, Haipeng Xiong, Ke Xian, Chen Zhao, Zhiguo Cao, and Xin Li. 2020. Sparse-to-Dense Depth Completion Revisited: Sampling Strategy and Graph Construction. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 682–699.

[27] J. Zhang, M. S. Ramanagopal, R. Vasudevan, and M. Johnson-Roberson. 2020. LiStereo: Generate Dense Depth Maps from LIDAR and Stereo Imagery. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 7829–7836. https://doi.org/10.1109/ICRA40945.2020.9196628

[28] J. Ziv and A. Lempel. 1977. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory* 23, 3 (1977), 337–343. https://doi.org/10.1109/TIT.1977.1055714