

# Semantics-aware Multi-modal Domain Translation: From LiDAR Point Clouds to Panoramic Color Images

Tiago Cortinhal  
Halmstad University  
Sweden

tiago.cortinhal@hh.se

Fatih Kurnaz  
Middle East Technical University  
Turkey

fatih@ceng.metu.edu.tr

Eren Erdal Aksoy  
Halmstad University  
Sweden

eren.aksoy@hh.se

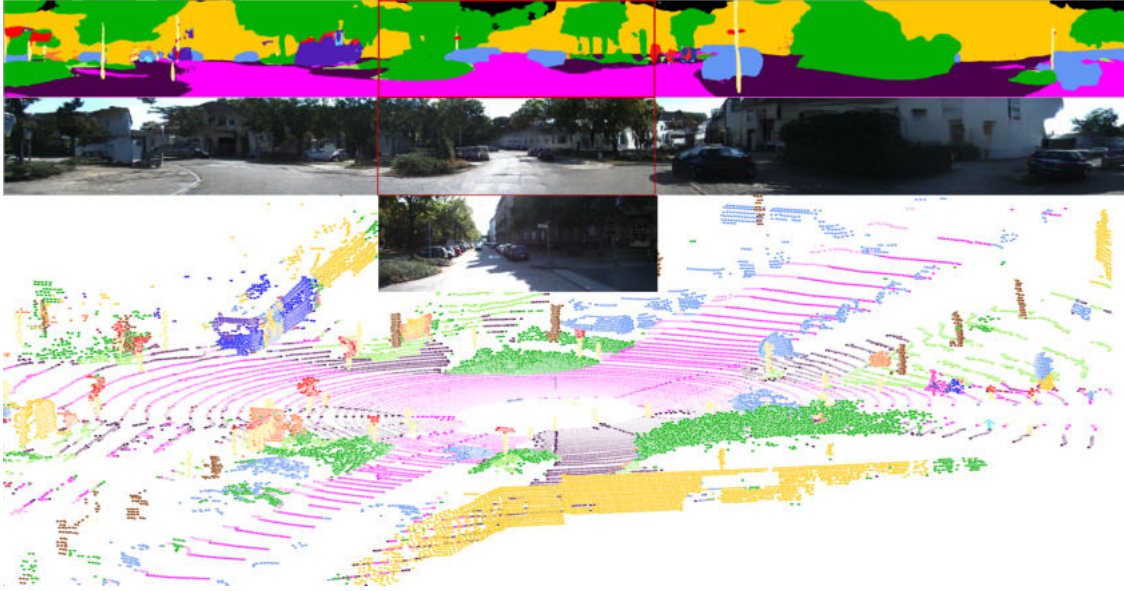


Figure 1: We propose a modular generative neural network framework that receives a full 3D LiDAR point cloud and returns the panoramic color image by solely relying on the semantics of the scene. The framework first applies semantic segmentation to the full LiDAR scan (the bottom image). Next, a novel generative network translates the LiDAR segments to the camera semantic segments (the top image), which are then converted back to the panoramic color images (the second image from the top) by an additional generative model. The red frame indicates the region that the ground-truth camera image (the third image from the top) corresponds to. Our framework, for the first time, generates a 360 degree color image of the environment.

## Abstract

*In this work, we present a simple yet effective framework to address the domain translation problem between different sensor modalities with unique data formats. By relying only on the semantics of the scene, our modular generative framework can, for the first time, synthesize a panoramic color image from a given full 3D LiDAR point cloud.*

*The framework starts with semantic segmentation of the point cloud, which is initially projected onto a spherical surface. The same semantic segmentation is applied to the corresponding camera image. Next, our new conditional*

*generative model adversarially learns to translate the predicted LiDAR segment maps to the camera image counterparts. Finally, generated image segments are processed to render the panoramic scene images. We provide a thorough quantitative evaluation on the Semantic-KITTI dataset [4] and show that our proposed framework outperforms other strong baseline models. Our source code is available.*

## 1. Introduction

Domain translation can be considered a mapping of data samples from an input source domain to a different target

domain. In computer vision and robotics, this subject has been vastly investigated to convert perceptual readings from one domain to another. For instance, translating sketches to images or segmentation maps to images, to name a few.

Although there exists an extensive literature on these kinds of image-to-image translations [12, 37, 33], recent works also focus on the multi-modal domain translation such as synthesizing images from raw 3D point sets [22, 3, 26]. The latter remains, unlike the former, relatively under-explored since point clouds, e.g. LiDAR scans, are sparse, unstructured, and nonuniformly sampled, which makes the mapping to the structured image space non-trivial.

Multi-modal domain translation has practical uses, in particular for autonomous vehicles. Take an example of having a failure in the camera setup. The lack of a modality can severely impair the autonomous vehicle’s performance since the subsequent sensor fusion and manoeuvre planning processes solely rely on these visual readings. Therefore, synthesizing photo-realistic images from other functioning modality readings, e.g. 3D LiDAR clouds, could help overcome a scenario of complete collapse. Another application could be generating additional annotated data in the source domain. By transferring the known labels across different domains, one can generate a new variation of the original scene from a different data distribution with no extra effort.

With this motivation, we propose a novel multi-modal domain translation framework leveraging the underlying semantics of the perceived scene. Differently from existing works [22, 15, 14], we argue that mediating the translation between perceptually different sensor readings via semantic scene segments could ease the process to a great extent.

**Our contribution:** More specifically, we propose a modular generative framework that can, for the first time, synthesize a panoramic color image from a full 3D LiDAR scan. See Fig. 11 for example. The framework, as shown in Fig. 2, starts with *SalsaNext* [7]: an off-the-shelf state-of-the-art model to semantically segment the point cloud, which is initially projected onto a spherical surface. The same semantic segmentation is applied to the paired camera image by employing another state-of-the-art model: SD-Net [30]. As our main technical contribution, we introduce a new conditional generative model, named TITAN-Net (generaTive domaIn TrANslation Network), which adversarially learns to translate the predicted LiDAR segment maps to the camera image counterparts. Finally, generated image segments are processed to render the panoramic scene images by a state-of-the-art model.

To the best of our knowledge, our framework is the first approach that relies on sensor-independent semantic context information to achieve semantically consistent translation between multi-modal domains. This opens a rich new vein of opportunity. We can handle possible camera failures by, for instance, rendering a raw 3D point set into an image.

Without any additional effort, we can further generate realistically looking variants of the scene image from the very same input point cloud. Hence, the available image datasets can simply be augmented with no extra cost.

We provide extensive quantitative and qualitative evaluations of our framework. Obtained results on the Semantic-KITTI dataset [4] show that our framework outperforms all evaluated strong baselines by a large margin.

## 2. Related Work

Although there is a large corpus of work on scene image generation [32, 19, 35] and point cloud rendering [25, 22, 3, 26, 15, 14, 29], studies that combine both are lackluster.

### 2.1. Image-to-Image Translation

Translating a scene image from one domain to another can be challenging but rewarding. Various methods [12, 37, 33] showed promising results in image to image translation. Some works [17, 28, 31, 24] also leveraged the image semantics to address the image translation and domain transfer problems. The same applies to image sequences where temporal cues need to be considered. For instance, [32] created the *vid2vid* network consisting of a carefully designed conditional Generative Adversarial Network (cGAN) with a spatio-temporal learning objective. The network can create temporally consistent and photo-realistic image sequences. Later the *vid2vid* model was improved in [19] by introducing memories of past frames to solve the long-term temporal consistency problem.

Furthermore, scene generation has also been addressed in the simulation domain. SurfGAN [35] uses texture-rich surfels to generate different trajectories from the same simulated environment.

### 2.2. Point Cloud Rendering

Most of the recent works converting point clouds to RGB images heavily rely on conditional GANs to force the network to generate more realistic images. In [25] the focus is on generating scene images from upsampled LiDAR data by using a simple cGAN, in which real images were used as the conditions for the cGAN. Other relevant works altered only the conditional part of the GAN. For example, [22] used predefined background image patches and viewpoint dependent projection to bias the network while generating images in compliance with 3D specifications. As an alternative, auxiliary conditional GAN was used in *pc2pix* [3] to render given a point cloud to its real-life shape from the desired camera angle while skipping the surface reconstruction process entirely. After applying arithmetic operations in the latent space, *pc2pix* can render various newly generated point clouds to images. Lastly, there are works [26] which employ cGAN for rendering point clouds without incorporating camera images in the rendering process.

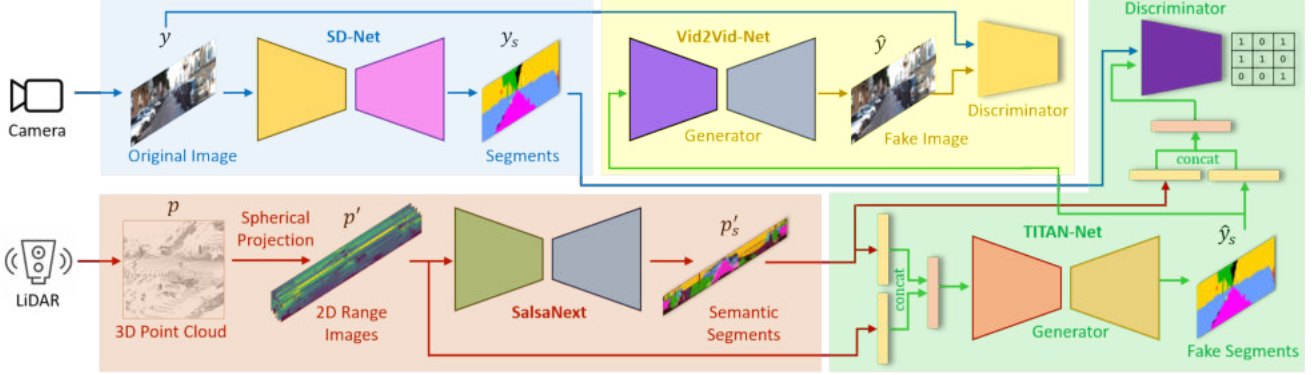


Figure 2: Our proposed modular framework has four neural networks. Each is depicted by a unique background color. In the red box, a captured 3D LiDAR point cloud  $p$  is first projected onto the 2D range image plane  $p'$  to be further processed by *SalsaNext* [7] to predict semantic segments  $p'_s$ . Likewise, the corresponding RGB camera image  $y$  is processed by SD-Net [30] to predict semantic segment maps  $y_s$  as depicted by the blue box. The green box highlights our proposed cGAN model TITAN-Net, where the *Generator* is conditioned on the concatenated  $p'_s$  and  $p'$  to generate the fake camera segment map  $\hat{y}_s$ . The TITAN-Net *Discriminator* is also conditioned on  $p'_s$  while comparing  $\hat{y}_s$  with the expected  $y_s$ . Finally, as depicted in the yellow box, the fake segment  $\hat{y}_s$  is processed by Vid2Vid-Net [32] to synthesize the realistic RGB image  $\hat{y}$ .

In addition to GANs, other generative models such as asymmetric encoder-decoder network [15] or U-NET [14] are also used to generate RGB images directly from point clouds. Similar to them, an end-to-end pipeline was developed in [29], which contains a point cloud encoder and RGB decoder in addition to a refinement network to improve photo quality. They can, thus, generate realistic RGB images from the novel viewpoints of a point cloud.

The closest works to ours are [22, 15, 14]. These approaches, however, neither have the capacity to process the full LiDAR point clouds nor exploit the scene semantics across modalities with different characteristics. Our framework differs in that the scene semantics acts as a bridge between the full-scan 3D point cloud and 2D image spaces to boost the domain translation.

### 3. Method

GANs [8] aim to learn a mapping from a noise vector  $z$  to a data type  $y$ , such that the *Generator* ( $G$ ) learns  $G_{z \rightarrow y} : z \rightarrow y$ . Conditional GANs [23], on the other hand, condition the generation of data solely on an additional vector of information  $s$  as  $G_{\{z, s\} \rightarrow y} : \{z, s\} \rightarrow y$ . The additional vector highly depends on the task at hand, but it can hold any feature set (e.g. sketches or semantic segments) from an image in a given domain to facilitate domain translation to a vector of classes expected to appear in the generated data.

Following the works of [12, 20], we omit the use of the noise vector  $z$  as the *Generator* will learn to ignore it and produce deterministic outputs. Nevertheless, several recent works on conditional generative models already addressed

the stochastic data generation [34, 10]. We focus only on the domain translation task with an intermediary representation between image and point cloud domains.

Given a LiDAR point cloud  $p \in \mathbb{R}^{n \times 4}$ , where  $n$  represents the number of points and each point has  $x, y, z$  coordinates and  $i$  intensity values, our goal is to generate an RGB image  $y \in \mathbb{R}^{w \times h \times 3}$  with a fixed image size:  $w$  (width) and  $h$  (height). We can define our domain translation as  $G_{\{p, s\} \rightarrow y} : \{p, s\} \rightarrow y$ , which is conditioned on the semantic segment maps  $s$ .

To solve this domain translation problem from  $p$  to  $y$ , we propose a modular framework consisting of four independently trained neural networks. Fig. 2 shows the overall framework. Note that our proposed solution is valid for  $G_{\{p, s\} \rightarrow y}$ , but is yet to be applied to  $G_{\{y, s\} \rightarrow p}$ . In the following, we give a detailed description of each network in the framework.

#### 3.1. LiDAR Point Cloud Segmentation

As depicted in the red box in Fig. 2, our framework starts with the semantic segmentation of 3D LiDAR point clouds.

To ease the correspondence problem between the unstructured point cloud  $p$  and structured image data  $y$ , we first apply a spherical projection [21, 2] to  $p$  and create the native LiDAR range view image  $p'$ . In this manner, each point in  $p$  is mapped to an image coordinate  $(u, v)$  as:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{1}{2}[1 - \arctan(y, x)\pi^{-1}]w' \\ [1 - (\arcsin(z, r^{-1}) + f_{down})f^{-1}]h' \end{pmatrix}, \quad (1)$$

where  $w'$  and  $h'$  denote the width and height of the projected image,  $r$  denotes the range of each point as  $r =$

$\sqrt{x^2 + y^2 + z^2}$  and  $f$  the vertical field of view as  $f = |f_{down}| + |f_{up}|$ . The final output of this transformation will be  $p' \in \mathbb{R}^{w' \times h' \times 5}$ , i.e. an image  $w' \times h'$  with  $(x, y, z, i, r)$  as channels. Note that the projection of  $p$  to  $p'$  does not lead to an enormous information loss in the point cloud since the depth information is still kept as an additional channel in  $p'$ .

The projected LiDAR data  $p'$  is then fed to an off-the-shelf semantic segmentation network *SalsaNext* [7] which has an encoder-decoder structure extended with an early context module capturing the global context information. The encoder unit consists of a stack of residual dilated convolution layers fusing receptive fields at various scales. The decoder part is composed of pixel-shuffle layers, which directly leverage the learned feature maps to upsample them with high accuracy and less computation. The final *SalsaNext* output is a 2D image  $p'_s \in \mathbb{R}^{w' \times h'}$  storing the predicted point-wise semantic segment labels.

### 3.2. Camera Image Segmentation

A similar segmentation treatment is also applied to the paired RGB camera images synchronized with the LiDAR point clouds as highlighted in the blue box in Fig. 2.

For this purpose, we employ another off-the-shelf state-of-the-art semantic segmentation network SD-Net [30] which receives the original RGB images  $y$  and returns the single-channel segment maps,  $y_s \in \mathbb{R}^{w \times h}$  with a fixed width ( $w$ ) and height ( $h$ ). SD-Net has a hierarchical attention architecture and learns to predict attention between adjacent scale pairs. Such multi-scale predictions are then combined at a pixel level to infer the semantic segments. SD-Net only operates during training, not in inference.

### 3.3. Translation of the LiDAR Semantic Segments

As highlighted in the green box in Fig. 2, the translation from 3D LiDAR point clouds to RGB camera images is triggered once both synchronized paired modality data, i.e.  $p$  and  $y$ , are represented by their corresponding semantic segments, i.e.  $p'_s$  and  $y_s$ . To convert the full-scan point cloud segments to their counterparts in the camera image space, we introduce a new semantics-aware conditional GAN model, named TITAN-Net.

**TITAN-Net:** Our cGAN involves two models: *Generator* and *Discriminator*. The *Generator* architecture is adapted from *SalsaNext* [7] since it is already designed to process range-view projected data, which is compatible with the input that TITAN-Net receives. In addition, *SalsaNext* has a lightweight model, thus, exhibits a high runtime performance (reaching up to 24 Hz) which allows fast TITAN-Net training. As shown in Fig. 2, the *Generator* receives as input both the range-view projections  $p'$  and the semantic segmentation maps  $p'_s$  coming from *SalsaNext* (see Sec. 3.1). To exploit both inputs more efficiently, we apply a  $1 \times 1$  convolution to each input before concatenating

them together. The merged inputs then pass through another  $1 \times 1$  convolutional layer. We further introduce a final up-sampling layer to the *SalsaNext* model to match the actual RGB image dimensions. Finally, the TITAN-Net *Generator* returns a fake camera image segment map,  $\hat{y}_s \in \mathbb{R}^{w \times h}$ . Note that LiDAR and camera have different fields of views of the same scene. To create the pairing between both during the training of TITAN-Net, we restrain the LiDAR projection to the approximate area corresponding to the scene in the camera image. The full range view image will be only used during the inference to create the panoramic images.

The TITAN-Net *Discriminator* network is based on the Pix2Pix *Discriminator*, commonly known as PatchGAN [12]. PatchGAN is an extension of [18] and assumes that the most relevant dependencies in an image are present at the patch level, usually called as Markov Random Fields (MRF). PatchGAN acts as a patch-wise classifier and outputs a 2D array corresponding to the image patches. This allows us to compute the *Discriminator* loss related to each region, assuming each non-overlapping area is independent. To compare with the *Generator* output, the *Discriminator* also receives the output of SD-Net (see Sec. 3.2), i.e.  $y_s$ , as the expected RGB image segmentation map. Like the TITAN-Net *Generator*, the *Discriminator* is also conditioned on the point cloud segments  $p'_s$ , and the same concatenation operation is applied before feeding the *Generator* output  $\hat{y}_s$  and  $p'_s$  to the *Discriminator*.

**Loss Function:** The TITAN-Net loss function is a linear combination of the Wasserstein GAN with Gradient Penalty (WGAN-GP) [9] ( $\mathcal{L}_{wgan-gp}$ ) and Lovász-Softmax [5] ( $\mathcal{L}_{ls}$ ) losses:  $\mathcal{L} = \mathcal{L}_{wgan-gp} + \mathcal{L}_{ls}$ .

As shown in [9], penalizing the gradient with WGAN-GP can stabilize the training procedure to reduce the mode collapse scenarios and ensure that a robust *Discriminator* can still pass relevant information back to the *Generator*. The WGAN-GP based *Discriminator* and *Generator* losses are defined as:

$$\mathcal{L}_D^{WGANGP} = \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [D(\tilde{x})] - \mathbb{E}_{x \sim \mathbb{P}_r} [D(x)] + \lambda \mathbb{E}_{\tilde{x} \sim \mathbb{P}_{\tilde{x}}} [(\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1)^2], \quad (2)$$

$$\mathcal{L}_G^{WGANGP} = -\mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [D(\tilde{x})], \quad (3)$$

where  $\mathbb{P}_r, \mathbb{P}_g, \mathbb{P}_{\tilde{x}}$  represent the real, generated, and sampling probabilities, respectively. The sampling probabilities are uniformly sampled along straight lines between  $\mathbb{P}_r$  and  $\mathbb{P}_g$  as in [9] and correspond to the gradient penalty.

We include the Lovász-Softmax [5] loss to directly optimize the Jaccard index, which is the main metric to evaluate the quality of semantic segments (see Sec. 4.3). Thus, during learning, we aim at maximizing the intersection-over-union score between the predicted and expected segments.

The term  $\mathcal{L}_{ls}$  acts as a guiding loss and is defined as:

$$\mathcal{L}_{ls} = \frac{1}{|C|} \sum_{c \in C} \overline{\Delta_{J_c}(m(c))}, \text{ and } m_i(c) = \begin{cases} 1 - x_i(c) & \text{if } c = y_i(c) \\ x_i(c) & \text{otherwise} \end{cases}, \quad (4)$$

where  $|C|$  represents the class number,  $\overline{\Delta_{J_c}}$  defines the Lovász extension of the Jaccard index,  $x_i(c) \in [0, 1]$  and  $y_i(c) \in \{-1, 1\}$  hold the predicted probability and ground truth label of pixel  $i$  for class  $c$ , respectively.

### 3.4. Segment to RGB Image Translation

As depicted in the yellow box in Fig. 2, our framework finally employs the camera segments  $\hat{y}_s$  generated by TITAN-Net to synthesize realistic RGB images  $\hat{y}$ .

For this purpose, we employ an off-the-shelf state-of-the-art cGAN model Vid2Vid-Net [32] which is coupled with spatio-temporal adversarial objectives to generate temporally consistent and photorealistic image sequences. Consequently, given the generated segment masks  $\hat{y}_s$ , Vid2Vid-Net returns the final response  $\hat{y}$  of our modular framework depicted in Fig. 2.

## 4. Experiments

### 4.1. Implementation Details

Except for Vid2Vid-Net, which is retrained using the default configurations from the available source code, we use the publicly available pre-trained weights for all other models in our proposed framework.

Regarding TITAN-Net, as an optimizer, we use Adam [16] with a learning rate of  $1 \times 10^{-4}$ ,  $(0.5, 0.999)$  as  $(\beta_1, \beta_2)$ . The batch size and dropout probability are fixed at 10 and 0.2, respectively. To avoid overfitting, we perform data augmentation by flipping randomly around the y-axis and randomly dropping points before creating the projection. Both augmentations are applied independently of each other with a probability of 0.5.

The entire proposed framework is trained with point clouds of size ranging from 10-13k points per scan and images of size  $1241 \times 376$ . After applying the spherical projections, we obtain the range view images with the size of  $2048 \times 64 \times 5$  centered on the view of the camera.

Our TITAN-Net model is implemented in PyTorch, and the source code is released for public use <https://github.com/Halmstad-University/TITAN-NET>.

### 4.2. Dataset

We evaluate the performance of the proposed framework and compare it to the other state-of-the-art domain translation approaches by using the large-scale challenging Semantic-KITTI dataset [4]. There exist over 43K point-wise annotated full 3D LiDAR scans and in total 19 different classes in the Semantic-KITTI dataset. By following the same protocol introduced in [21], we divide the dataset into

training (sequences 00-10), validation (sequence 08), and test splits (sequences 11-21).

Note that although *SalsaNext* is already trained on the Semantic-KITTI dataset, the SD-Net model is trained using the Cityscapes dataset [6] which has fewer classes, i.e. 14, with different labels. To cope with the incompatibilities between the class numbers and labels in two datasets, we define a mapping table (see the supplementary material) that returns 14 unique class labels matched in both datasets.

### 4.3. Evaluation Metrics

We use the following metrics to measure the quality of the generated segment maps and synthesized images:

**Jaccard Index:** For the quality evaluation of the predicted segment maps, we use the Jaccard Index, a.k.a. the mean intersection-over-union (mIoU), over all classes. A higher mIoU score indicates better segmentation results.

**Structural Similarity Index Measure (SSIM):** SSIM is a perception metric that measures image similarity by exploiting three different image components: Luminance, Contrast, and Structure. Both images are normalized, and we compare the covariance of both images. In our evaluations, we use SSIM with a window size of 11 as in the original paper [36]. The higher the SSIM value, the better.

**Fréchet Inception Distance (FID):** The lower the FID value, the closer is the generated image to the real counterpart. As described in [11], FID is consistent with human judgment. Any noise or artifacts present in the generated image decrease the FID value. Generally, FID is a reliable metric as it correlates consistently with the visual quality of generated images.

**Sliced Wasserstein Distance (SWD):** Another metric allowing us to weigh the distribution of generated and real images is the Sliced Wasserstein Distance (SWD) proposed in [13]. This metric assumes that a successful *Generator* produces images that have structural similarities at different scales. We extract image patches from a Laplacian Pyramid [1] starting with a low-pass resolution of  $16 \times 16$ , which is doubled until the desired resolution is reached.

After the respective normalization w.r.t. the mean and standard deviation, the SWD value between both sets of patches is computed following the work in [27]. Generally, the lower the SWD value, the better.

### 4.4. Baselines

We compare the performance of our framework to two different baselines trained on the same dataset using the same training protocol.

**Pix2Pix** [12] is the state-of-the-art generative model for image-to-image translation. In our framework in Fig. 2, we replace TITAN-Net with Pix2Pix to diagnose the contribution of our generative model TITAN-Net in the generated segment maps and synthesized images.

Approach	Car	Bicycle	Motorcycle	Truck	Other-Vehicle	Person	Road	Sidewalk	Building	Fence	Vegetation	Terrain	Pole	Traffic-Sign	mIoU $\uparrow$
Pix2Pix [12]	8.8	0	0	0	0	0	57.7	15.7	32.8	12.5	32.7	14.8	0.5	0	12.5
TITAN-Net (Ours)	<b>68.2</b>	<b>9.9</b>	<b>7.6</b>	<b>7.6</b>	0	<b>7.3</b>	<b>75.4</b>	<b>48.3</b>	<b>62.9</b>	<b>33.6</b>	<b>60.1</b>	<b>49.9</b>	<b>2</b>	<b>3.7</b>	<b>31.1</b>

Table 1: Quantitative results for the generated semantic segment images on the test sequences.  $\uparrow$  denotes that higher is better.

Approach	SWD $\times 10^3 \downarrow$									
	SSIM $\uparrow$	FID $\downarrow$	1024 $\times$ 1024	512 $\times$ 512	256 $\times$ 256	128 $\times$ 128	64 $\times$ 64	32 $\times$ 32	16 $\times$ 16	avg
SC-UNET $\sim$	0.3158	261.282	2.65	2.56	2.36	2.20	2.14	2.14	4.07	2.59
Pix2Pix $\sim \rightarrow$ Vid2Vid	0.2543	73.476	2.29	2.22	2.18	2.15	2.11	<b>2.13</b>	3.95	2.59
TITAN-Net (Ours) $\rightarrow$ Vid2Vid	0.2610	<b>61.914</b>	<b>2.22</b>	<b>2.18</b>	<b>2.15</b>	<b>2.11</b>	<b>2.08</b>	<b>2.13</b>	<b>3.78</b>	<b>2.38</b>
Pix2Pix $\sim$ - w/o SegMap	0.2006	209.150	2.43	2.35	2.31	2.31	2.35	2.42	4.76	2.71
TITAN-Net - w/o SegMap	<b>0.3692</b>	326.298	3.24	3.14	2.98	2.61	2.23	2.10	3.94	2.89
TITAN-Net - w/o Rangeview $\rightarrow$ Vid2Vid	0.2442	76.932	2.31	2.23	2.18	2.16	2.15	2.22	4.47	2.53
SD-Net -> Vid2Vid	0.4089	20.3694	2.10	1.99	1.86	1.71	1.56	1.41	1.70	1.76

Table 2: Quantitative results for the synthesized RGB images using the test sequences. Each level of the Laplacian Pyramid corresponds to a given resolution. The distances are shown per level and the average gives us the overall distance between both distributions. Due to the nature of SWD, both images are resized to  $1024 \times 1024$  before calculating the distance.  $\downarrow$  denotes that lower scores are better and  $\uparrow$  that higher is better.

**SC-UNET** [14] is a recent generative model based on Selected Connection U-Net (SC-UNET) specifically designed for generating RGB images directly from point clouds. Unlike our approach, SC-UNET neither incorporates the segment maps nor involves adversarial training.

#### 4.5. Quantitative & Qualitative Results

We start with evaluating the quality of the generated semantic segmentation masks since the remaining image synthesis solely relies on these segments in our proposed framework. Table 1 shows the obtained mIoU scores on the validation set for the segment maps  $\hat{y}_s$  generated from  $p'$  and  $p'_s$  (see Fig. 2). Note that the output of SD-Net [19], i.e.  $y_s$ , is here considered as the ground-truth since it is also employed by the TITAN-Net and Pix2Pix *Discriminators*. Table 1 shows that the replacement of our proposed TITAN-Net model with the Pix2Pix [12] counterpart in Fig. 2 leads to a substantial drop in the segmentation accuracy, without having any exception in the individual classes.

Obtained quantitative results on the quality of the final synthesized RGB images,  $\hat{y}$ , are reported in Table 2. We, here, compare the performance of TITAN-Net combined with Vid2Vid (i.e. TITAN-Net  $\rightarrow$  Vid2Vid) to the other approaches, i.e. SC-UNET [14] and Pix2Pix [12] combined with Vid2Vid (i.e. Pix2Pix  $\rightarrow$  Vid2Vid). Table 2 clearly shows that our proposed approach (i.e. TITAN-Net  $\rightarrow$  Vid2Vid) considerably outperforms the others by leading to the lowest FID and SWD scores.

When it comes to the SSIM metric, SC-UNET [14] performs better than the other two methods. We will elaborate

more on this result in section 4.6.

Fig. 3 shows sample segment maps and RGB images generated by our framework (i.e. TITAN-Net  $\rightarrow$  Vid2Vid) in comparison with Pix2Pix [12] and Vid2Vid (i.e. Pix2Pix  $\rightarrow$  Vid2Vid). This figure clearly shows that our TITAN-Net model can reconstruct more accurate segment maps, thus, has much better image synthesis capability compared to Pix2Pix. For instance, the semantically important classes (such as buildings, roads, and vehicles) are reconstructed with high fidelity, as depicted in Fig. 3. Note that since SC-UNET [14] does not rely on the segment masks, it is omitted in this figure.

#### 4.6. Ablation Study

We conduct ablation studies to better understand the contribution of different components in our framework.

**Effect of the semantic segmentation maps:** We assess the contribution of semantic segments to the image reconstruction process. Therefore, we measured the performance of TITAN-Net and Pix2Pix when the intermediate semantic segmentation step is completely bypassed as in the case of SC-UNET [14]. Here, the RGB images  $\hat{y}$  are directly recovered from the projected point cloud images  $p'$ . We followed the same training protocol defined for the previous experiments. However, we used Mean Squared Error (MSE) as our guiding loss in Eq. 4 since the *Lovász-Softmax* loss only applies to segments.

Obtained results without accessing the segmentation maps (w/o *SegMap*) for both TITAN-Net and Pix2Pix are respectively reported in the fourth and fifth rows in Table 2.

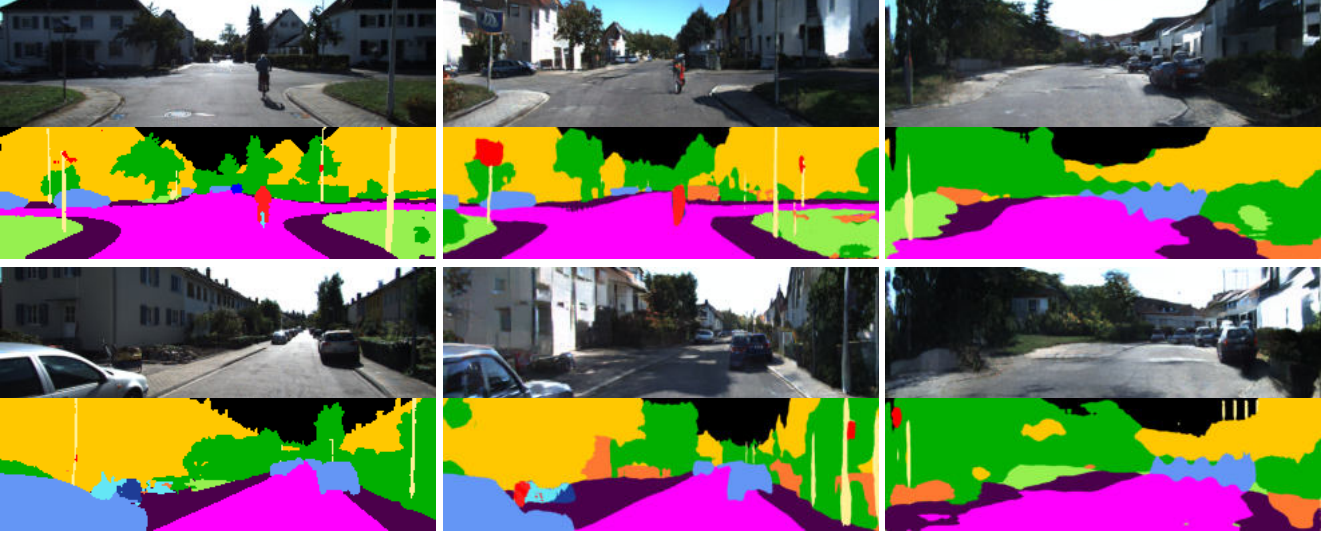


Figure 3: Sample qualitative results are showing the synthesized images at the top together with the corresponding generated segment maps at the bottom. From left to right, we have the ground-truth images, the TITAN-Net results (Ours) and the Pix2Pix [12] outputs. Note that TITAN-Net and Pix2Pix are combined with Vid2Vid to translate segments to RGB images.



Figure 4: Sample images generated directly from the projected point cloud images without employing the segmentation maps. From left to right we have the ground-truth image and the results from TITAN-Net (Ours), Pix2Pix [12], and SC-UNET [14].

Qualitative synthesized images are also depicted in Fig. 4. These results convey the fact that excluding the semantics drastically diminishes the quality of synthesized images. The reason for obtaining a better SSIM score in Table 2 is that the model rather learns how the luminance and contrast behave but fails to capture abstract context information as shown in Fig. 4.

**Effect of having  $p'$  as a condition in TITAN-Net:** In the previous ablation study, we implicitly investigated the role of LiDAR segments  $p'_s$  in image synthesis. We now diagnose the contribution of  $p'$  in the quality of generated RGB images. The last row in Table 2 shows that when the condition on  $p'$  is removed, the obtained results get worse in contrast to the results in the third row.

#### 4.7. Runtime

The runtime for training on two Quadros RTX 6000 GPUs is about six days for 19K training samples and 118 epochs. Regarding the inference time, on a single Quadros RTX 6000 GPU it takes 40.79, 198.75, 42.15, 123.72 msecs for point cloud segmentation, image segmentation, translation between segment maps, and synthesizing a  $376 \times 1241$

image. When it comes to our baseline models, translation between segment maps takes about 22.48 and 87.72 msecs for the Pix2Pix [12] and SC-UNET [14] models, respectively. These values were calculated on the validation split.

## 5. Limitations and Discussion

In this work, we argue that employing scene semantics is of utmost importance in translating features between the domains that have unique data formats such as 3D point clouds and 2D images. Findings provided in Table 2 and Figs. 3-4 clearly support our hypothesis that the semantic data representation can, to a great extent, alleviate the domain translation problem across different sensor modalities.

Unlike other relevant works [22, 15, 14], our proposed framework has the capacity to process the full  $360^\circ$  LiDAR scan. This gives us a unique chance to synthesize panoramic camera images from the projected point cloud range images  $p'$ . Fig. 5 illustrates two sample panoramic images generated by our framework using the test split of the dataset. Note that due to lack of ground truth, it is non-trivial to evaluate the quality of these rendered panoramic images. Therefore, the results presented so far in Figs. 3-4 involve images



Figure 5: Two sample panoramic images synthesized by our proposed TITAN-Net model on the Semantic-KITTI test set.

synthesized only from a restricted region in the LiDAR projection, which approximates the original camera view, i.e. the only available ground truth. We emphasize that this novel contribution plays a crucial role in handling possible sensor failures in autonomous vehicles. Take an example of having a failed camera sensor. The missing scene images can then be translated from other functional sensors, e.g. LiDAR. Thus, the vehicle can employ these generated images as *initial beliefs* to bootstrap the subsequent sensor fusion and maneuver planning processes instead of simply having a sudden emergency stop.

Furthermore, such a smooth translation between different modalities can allow us to gather, for instance, additional annotated data with no extra effort. The top row in Fig 6 shows an original camera image from the Semantic-KITTI dataset. By using the corresponding LiDAR scan,



Figure 6: Different variants generated by our framework. From top to bottom, they are the ground truth camera image, generated image by TITAN-Net → Vid2Vid trained on the Semantic-KITTI dataset, and synthesized image by TITAN-Net → Vid2Vid trained on Cityscapes, respectively.

our framework trained on this dataset can already produce a variation of this scene as depicted in the middle row in Fig 6. We can now simply replace the Vid2Vid head with the version trained on the Cityscapes dataset [6] to produce a different variant as shown in the bottom row in Fig 6. Producing such different variants without additional effort can help us to augment the available image datasets needed to efficiently regularize the neural networks.

We are aware of the fact that some vehicle samples in the generated images may have visual artifacts, e.g. vehicle boundaries are not preserved, in particular, when the scene has multiple vehicle samples (see Fig. 5). The main reason is that our framework relies on *SalsaNext* and SD-Net, which are not instance-aware segmentation approaches. We believe that segmenting individual instances can largely mitigate this problem. Thanks to having a modular framework, the segmentation networks can easily be replaced with the instance-aware counterparts in Fig. 2.

Another limitation in our framework is that each LiDAR and camera data is treated individually. Thus there is no temporal consistency between synthesized images. We plan to extend our approach by incorporating temporal cues to overcome this issue.

In the supplementary material, we provide more images together with a video<sup>1</sup> showing the performance of TITAN-Net on the validation and test splits.

## 6. Conclusion

In this work<sup>2</sup>, we introduce a novel semantics-aware domain translation framework to synthesize panoramic color images from a given point cloud. Our framework is a modular approach and involves four different models trained individually. The framework relies on our new cGAN model, TITAN-Net, which translates the LiDAR semantic maps to camera image formats to boost the image generation.

<sup>1</sup><https://youtu.be/eV510t29TAc>

<sup>2</sup>The research leading to these results has received funding from the Vinnova FFI project SHARPEN, under grant agreement no. 2018-05001.

## References

- [1] Edward Adelson, Charles Anderson, James Bergen, Peter Burt, and Joan Ogden. Pyramid methods in image processing. *RCA Eng.*, 29, 11 1983. [5](#)
- [2] Eren Erdal Aksoy, Saimir Baci, and Selcuk Cavdar. Salsanet: Fast road and vehicle segmentation in lidar point clouds for autonomous driving. In *IEEE IV*, 2020. [3](#)
- [3] Rowel Atienza. A conditional generative adversarial network for rendering point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. [2](#)
- [4] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Juergen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *ICCV*, 2019. [1](#), [2](#), [5](#)
- [5] M. Berman, A. Triki, and Matthew B. Blaschko. The lovasz-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, 2018. [4](#)
- [6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [5](#), [8](#)
- [7] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanet: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In *International Symposium on Visual Computing*, pages 207–222. Springer International Publishing, 2020. [2](#), [3](#), [4](#)
- [8] Ian J. Goodfellow, Jean Pouget-Abadie, M. Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. [3](#)
- [9] Ishaan Gulrajani, F. Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In *NIPS*, 2017. [4](#)
- [10] Yang He, Bernt Schiele, and Mario Fritz. Diverse conditional image generation by stochastic regression with latent drop-out codes. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 422–437, Cham, 2018. Springer International Publishing. [3](#)
- [11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017. [5](#)
- [12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017. [2](#), [3](#), [4](#), [5](#), [6](#), [7](#)
- [13] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. [5](#)
- [14] Hyun-Koo Kim, K. Yoo, and Ho-Youl Jung. Color image generation from lidar reflection data by using selected connection unet. *Sensors (Basel, Switzerland)*, 20, 2020. [2](#), [3](#), [6](#), [7](#)
- [15] Hyun-Koo Kim, Kook-Yeol Yoo, Ju H. Park, and Ho-Youl Jung. Asymmetric encoder-decoder structured fcn based lidar to color image generation. *Sensors*, 19(21):4818, Nov. 2019. [2](#), [3](#), [7](#)
- [16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. [5](#)
- [17] Samuel Lavoie-Marchildon, Faruk Ahmed, and Aaron Courville. Integrating categorical semantics into unsupervised domain translation. In *International Conference on Learning Representations*, 2021. [2](#)
- [18] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 702–716, Cham, 2016. Springer International Publishing. [4](#)
- [19] Arun Mallya, Ting-Chun Wang, Karan Sapra, and Ming-Yu Liu. World-consistent video-to-video synthesis. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 359–378, Cham, 2020. Springer International Publishing. [2](#), [6](#)
- [20] Michaël Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016. [3](#)
- [21] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. Rangenet ++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4213–4220, 2019. [3](#), [5](#)
- [22] S. Milz, M. Simon, K. Fischer, Maximilian Pöpperl, and H. Groß. Points2pix: 3d point-cloud to image translation using conditional gans. In *Gcpr*, 2019. [2](#), [3](#), [7](#)
- [23] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. *arXiv e-prints*, page arXiv:1411.1784, Nov. 2014. [3](#)
- [24] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Instagan: instance-aware image-to-image translation. In *International Conference on Learning Representations*, 2019. [2](#)
- [25] Zhenchao Ouyang, Yu Liu, C. Zhang, and J. Niu. A cgans-based scene reconstruction model using lidar point cloud. *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, pages 1107–1114, 2017. [2](#)
- [26] T. Peters and C. Brenner. Conditional adversarial networks for multimodal photo-realistic point cloud rendering. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, pages 1–13, 2020. [2](#)
- [27] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In Alfred M. Bruckstein, Bart M. ter Haar Romeny, Alexander M. Bronstein, and Michael M. Bronstein, editors, *Scale Space and Variational Methods in Computer Vision*, pages 435–446, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. [5](#)

- [28] Pravekar Roy, Nicolai Häni, and Volkan Isler. Semantics-aware image to image translation and domain transfer. *CoRR*, 2019. [2](#)
- [29] Zhenbo Song, Wayne Chen, Dylan Campbell, and Hongdong Li. Deep novel view synthesis from colored 3d point clouds. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 1–17, Cham, 2020. Springer International Publishing. [2](#), [3](#)
- [30] Andrew Tao, Karan Sapra, and Bryan Catanzaro. Hierarchical multi-scale attention for semantic segmentation. *CoRR*, 2020. [2](#), [3](#), [4](#)
- [31] Matteo Tomei, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Art2real: Unfolding the reality of artworks via semantically-aware image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. [2](#)
- [32] T. Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, A. Tao, J. Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NeurIPS*, 2018. [2](#), [3](#), [5](#)
- [33] T. Wang, Ming-Yu Liu, Jun-Yan Zhu, A. Tao, J. Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018. [2](#)
- [34] Ze Wang, X. Cheng, G. Sapiro, and Qiang Qiu. Stochastic conditional generative networks with basis decomposition. In *ICLR*, 2020. [3](#)
- [35] Zhenpei Yang, Y. Chai, Dragomir Anguelov, Y. Zhou, Pei Sun, D. Erhan, S. Rafferty, and Henrik Kretschmar. Surfelgan: Synthesizing realistic sensor data for autonomous driving. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11115–11124, 2020. [2](#)
- [36] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. [5](#)
- [37] Jun-Yan Zhu, T. Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017. [2](#)

# Supplementary Material

## Semantics-aware Multi-modal Domain Translation: From LiDAR Point Clouds to Panoramic Color Images

We here provide more additional material to support our main submission. In the first section, we provide a detailed description of the TITAN-Net *Generator* and *Discriminator* architectures. Next, we provide a table showing the corresponding label matches between the Semantic-KITTI and Cityscapes datasets. We also present an additional ablation study. Finally, we provide more qualitative experimental results, e.g. more synthesized images and videos showing the performance of TITAN-Net on the validation and test splits.

### 1. TITAN-Net Architecture

As described in the main manuscript TITAN-Net is a conditional GAN model involving two units: *Generator* and *Discriminator*.

**TITAN-Net Generator:** Fig. 1 shows the *Generator* architecture, which is divided into four main components, described as follows:

- **Merge Module:** This module is responsible for the early fusion of LiDAR semantic segmentation maps and range-view projections. Instead of a naive concatenation of both inputs, we, first, feed each of them to a  $1 \times 1$  convolutional layer before concatenating and feeding to another  $1 \times 1$  convolutional layer. This allows us to exploit the raw inputs more efficiently. The choice of  $1 \times 1$  kernel also comes from the fact that we pretend to combine the inputs from a local perspective without considering the surrounding pixels.
- **Contextual Module:** The *Generator* has a contextual module which learns the global context information, e.g. complex correlations between segment classes by large receptive fields. More precisely, the contextual module has a set of residual dilated convolution operations fusing a large receptive field ( $3 \times 3$ ) with a smaller one ( $1 \times 1$ ) through a skip connection to aggregate the context cues in different regions. This way, the *Generator* captures fine detailed spatial information while extracting the global context.
- **U-Net architecture:** The main skeleton of the *Generator* relies on a U-Net like encoder-decoder architec-

ture. The encoder unit has blocks of dilated convolutions (see Block I in Fig. 1) with gradually increasing receptive fields of  $3 \times 3$ ,  $5 \times 5$ , and  $7 \times 7$ . Each block first passes the received feature map through a  $1 \times 1$  layer that will be the primary skip connection after the inner block. The decoder employs *pixel-shuffle* layers (see Block III in Fig. 1) that exploit the learnt feature maps to upsample the spatial dimension. Unlike conventional transpose convolutions which are prone to checkerboard artifacts, *pixel-shuffle* layers have less parameters and force the learnt feature maps to retain more information. The decoder involves another blocks of dilated convolutions (Block V in Fig. 1) to extract more descriptive features.

- **Output:** As shown in Fig. 1, we finally perform a bilinear upsampling at the end of the network to generate segmentation maps in the camera image space while avoiding visual artifacts from the start.

**TITAN-Net Discriminator:** The *Discriminator* follows the same central idea as in the *Generator*. Both inputs are fused using a convolutional block as illustrated in Fig. 2. In this case, the conditional input (the LiDAR segmentation map) is upsampled to meet the original camera image dimension before the concatenation.

The *Discriminator* model follows the PatchGAN structure. This type of *Discriminator* focuses on penalising structures at a local patch scale. Instead of mapping an entire image to a single scalar, we have a final and smaller representation of the inputs, representing the realism of the original image's different and independent regions. This can be seen as if we manually divide the original image into smaller patches and pass each of them through the *Discriminator*. The advantage of this method relies on the fact that we do not need to apply a preprocessing to the inputs - to divide them into patches - while the *Generator* is still able to operate on the full input instead of patches that can hurt the performance.

As depicted in Fig. 2, the TITAN-Net *Discriminator* has six strided convolutional blocks that yields a final patch size of  $10 \times 3$ . No normalisation is used.

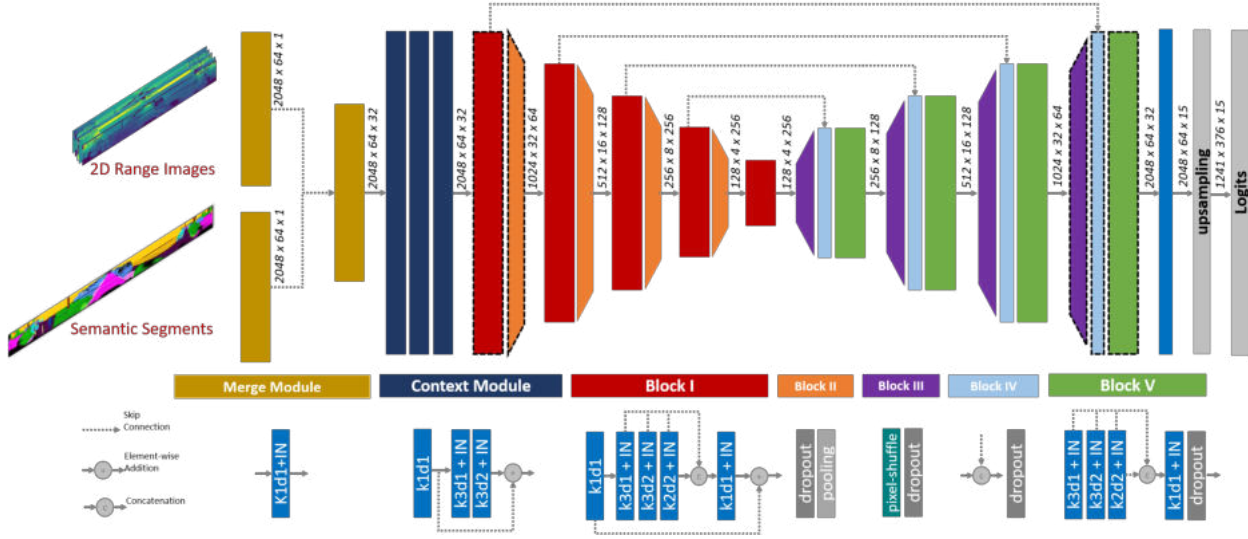


Figure 1: TITAN-Net Generator. Blocks with dashed edges are not involving dropout. The abbreviations  $k$ ,  $d$ , and  $IN$  stand for the kernel size, dilation rate and instance normalization, respectively.

## 2. Mapping Between Different Datasets

As already described in the main manuscript, *Sal-saNext* and SD-Net are trained on the Semantic-KITTI and Cityscapes datasets, respectively. Both dataset have different classes with unique class labels. To cope with the incompatibilities between these differences in two datasets, we define Table 1 returning 14 unique class labels matched in both datasets. This mapping allows us to have a better alignment between LiDAR and RGB labels used for training of TITAN-Net.

## 3. Ablation Study

As described in the main manuscript, the final TITAN-Net loss function has two components: Wasserstein GAN with Gradient Penalty ( $\mathcal{L}_{wgan-gp}$ ) and *Lovász-Softmax* ( $\mathcal{L}_{ls}$ ). We, here, ablate the *Lovász-Softmax* loss term to diagnose the overall contribution in the network performance. As reported in the last row of Table 2, we observe that the term  $\mathcal{L}_{ls}$  has a certain contribution to the generation of the semantic segment maps. Including both loss terms lead to the best results (see the second row in Table 2), which sug-

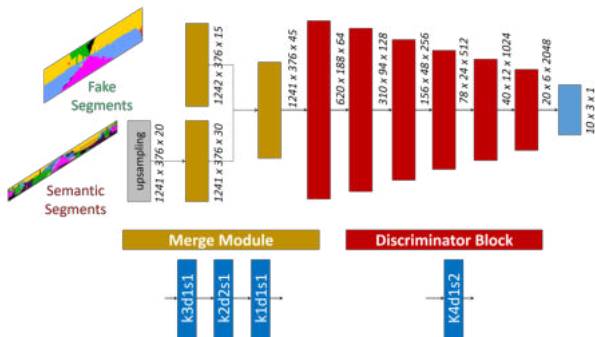


Figure 2: TITAN-Net Discriminator. The abbreviations  $k$  and  $d$  stand for the kernel size and dilation rate, respectively.

		Cityscapes																
		Unlabeled	Car	Bicycle	Motorcycle	Truck	Other-Vehicle	Person	Road	Sidewalk	Building	Fence	Vegetation	Terrain	Pole	Traffic-Sign		
SemanticKITTI	Unlabeled	X																
	Car		X															
	Bicycle			X														
	Motorcycle				X													
	Truck					X												
	Other-Vehicle						X											
	Person							X										
	Bicyclist							X										
	Motorcyclist							X										
	Road								X									
	Parking									X								
	Sidewalk										X							
	Other-Ground											X						
	Building												X					
	Fence													X				
	Vegetation														X			
	Trunk															X		
Terrain																X		
Pole																	X	
Traffic-Sign																		X

Table 1: Mapping between the labels available on the Cityscapes and SemanticKITTI datasets.

Approach	Car	Bicycle	Motorcycle	Truck	Other-Vehicle	Person	Road	Sidewalk	Building	Fence	Vegetation	Terrain	Pole	Traffic-Sign	mIoU $\uparrow$
Pix2Pix	8.8	0	0	0	0	0	57.7	15.7	32.8	12.5	32.7	14.8	0.5	0	12.5
TITAN-Net (Ours)	<b>68.2</b>	<b>9.9</b>	<b>7.6</b>	<b>7.6</b>	0	<b>7.3</b>	75.4	<b>48.3</b>	<b>62.9</b>	<b>33.6</b>	<b>60.1</b>	<b>49.9</b>	<b>2</b>	<b>3.7</b>	<b>31.1</b>
TITAN-Net (w/o $\mathcal{L}_{ls}$ )	60.5	0	0	0.3	0	0.05	<b>78.0</b>	43.6	56.9	33.0	52.8	42.9	0.08	0.07	26.2

Table 2: Quantitative results for the generated semantic segment images on the test sequences.  $\uparrow$  denotes that higher is better.

gests that both losses regularize the network in a complementary manner.

## 4. Qualitative Results

In the following, we provide more qualitative results compared to the other baseline models on the test dataset. The following figures show different qualitative results:

- Fig. 3 shows sample segment maps and RGB images generated by our framework (i.e. TITAN-Net  $\rightarrow$  Vid2Vid) in comparison with Pix2Pix and Vid2Vid (i.e. Pix2Pix  $\rightarrow$  Vid2Vid),
- Fig. 4 presents sample images synthesized directly from the raw projected point cloud images without employing the segmentation maps.
- Fig. 5 shows sample panoramic images synthesized by our TITAN-Net model on the Semantic-KITTI test set.
- Fig. 6 depicts different variants of the original camera image, generated by our framework (i.e. TITAN-Net  $\rightarrow$  Vid2Vid) using the Semantic-KITTI and Cityscapes datasets.

## 5. Video

We also provide three videos showing the performance of TITAN-Net on the validation and test splits of the Semantic-KITTI dataset. All three videos are available<sup>1 2 3</sup>. Note that in our proposed framework each LiDAR and camera data is treated individually. Thus, there is no temporal consistency between synthesized images.

<sup>1</sup><https://youtu.be/He6fKkF88IE>

<sup>2</sup><https://youtu.be/k59zmVhsKVI>

<sup>3</sup><https://youtu.be/zR6Ix6YUhwI>

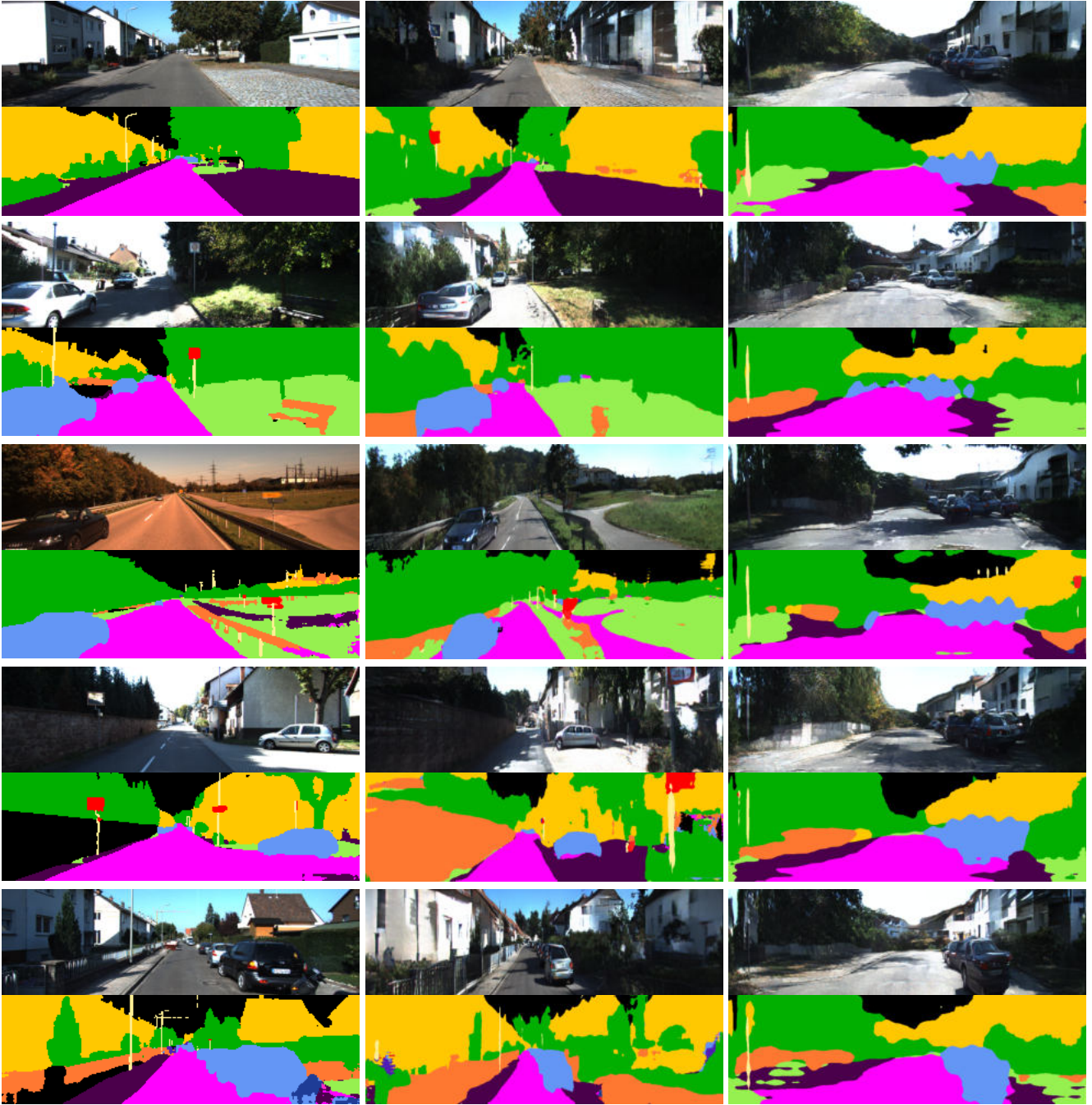


Figure 3: Qualitative results on the Semantic-KITTI test set. The synthesized images are shown at the top and the corresponding generated segment maps are depicted at the bottom. From left to right, we have the ground-truth images, the TITAN-Net results (Ours) and the Pix2Pix outputs. Note that TITAN-Net and Pix2Pix are combined with Vid2Vid to translate segments to RGB images.



Figure 4: Sample images generated directly from the projected point cloud images without employing the segmentation maps. From left to right we have the ground-truth image and the results from TITAN-Net (Ours), Pix2Pix, and SC-UNET.



Figure 5: Panoramic images synthesized by our proposed TITAN-Net model on the Semantic-KITTI test set.

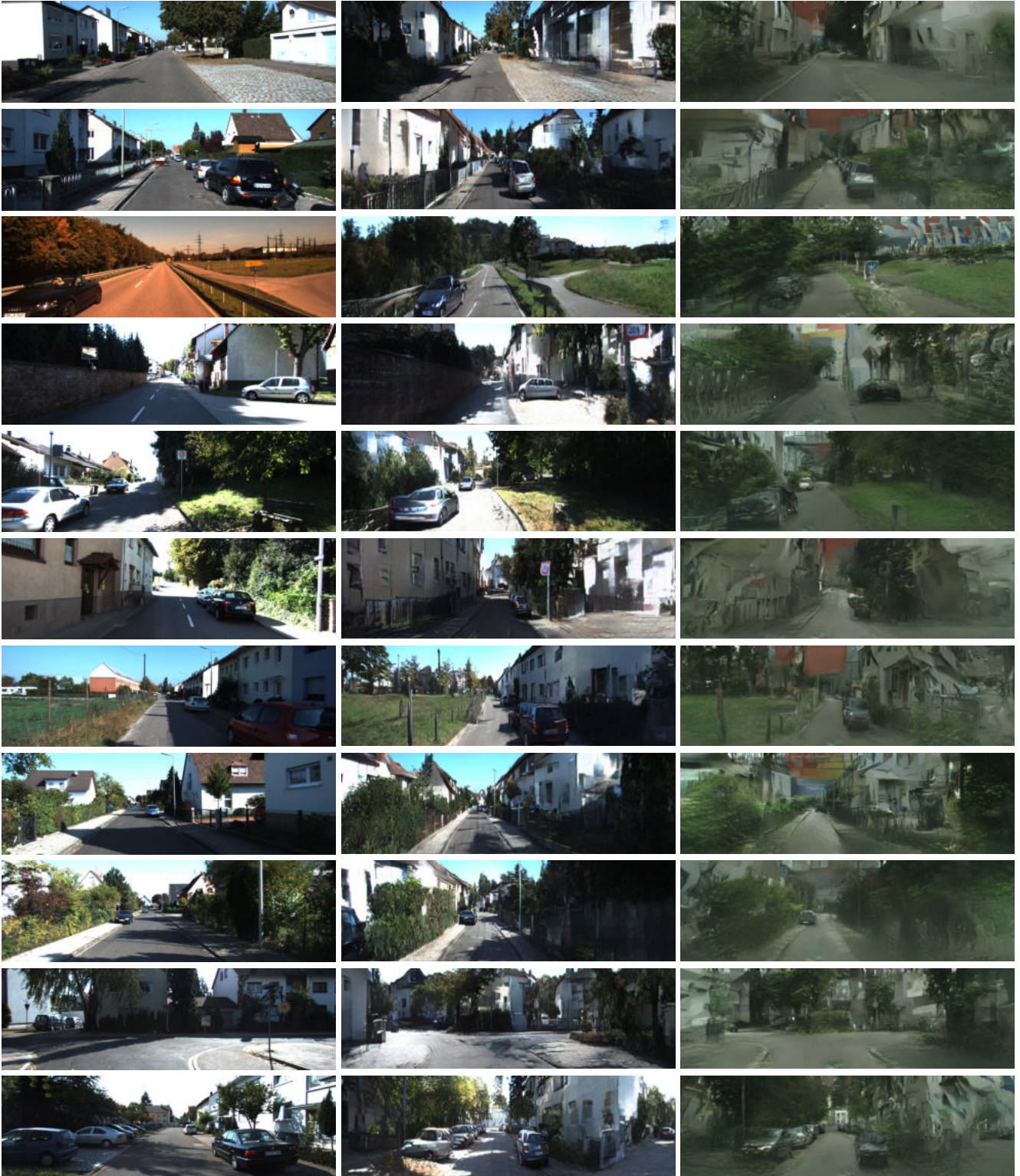


Figure 6: Different variants generated by our framework on the test set. From left to right, they are the ground truth camera image, generated image by TITAN-Net  $\rightarrow$  Vid2Vid trained on the Semantic-KITTI dataset, and synthesized image by TITAN-Net  $\rightarrow$  Vid2Vid trained on Cityscapes, respectively.