Depth Completion Auto-Encoder

Kaiyue Lu[1,2], Nick Barnes[1], Saeed Anwar[2,1], and Liang Zheng[1]

[1]Australian National University [2]Data61, CSIRO

Abstract

This paper proposes a new approach to integrating image features for unsupervised depth completion. Instead of resorting to the image as input like existing works, we propose to employ the image to guide the learning process. Specifically, we regard dense depth as a reconstructed result of the sparse input, and formulate our model as an auto-encoder. To reduce structure inconsistency resulting from sparse depth, we employ the image to guide latent features by penalizing their difference in the training process. The image guidance loss enables our model to acquire more dense and structural cues that are beneficial for producing more accurate and consistent depth values. For inference, our model only takes sparse depth as input and no image is required. Our paradigm is new and pushes unsupervised depth completion further than existing works that require the image at test time. On the KITTI Depth Completion Benchmark, we validate its effectiveness through extensive experiments and achieve promising performance compared with other unsupervised works. The proposed method is also applicable to indoor scenes such as NYUv2.

1. Introduction

Unsupervised depth completion aims to recover dense depth from the sparse input without the supervision of dense ground truth. Compared with the supervised setting, unsupervised models do not involve expensive manual annotation.

In the depth completion community, a commonly acknowledged challenge is structure inconsistency, *i.e.*, object structures cannot be correctly identified and recovered [30, 35, 9]. Essentially, this problem is caused by the sparse nature of the input, *e.g.*, we can hardly tell where the car boundary is in Fig. 1(b) due to too many missing depth values. Fully-supervised models can reduce structure inconsistency by making use of dense ground truth, which provides pixel-wise supervision and covers most object structures. Many supervised works also address this problem by taking the RGB image as an extra input and fusing image-inspired structural features with sparse depth either through early or late fusion [35, 46, 7, 22].

For unsupervised depth completion, structure inconsistency becomes even harder to overcome because there is no dense ground truth available. Among the few works in the unsupervised setting, traditional non-learning methods[1] [24, 40, 3, 12] use hand-crafted matrix interpolation operations to fill in missing values, but lack effective image guidance. Recently, an alternative practice is to make use of network training, and take the RGB image as an additional input and calculate the image warping loss either from stereo [55] or adjacent video frames [32, 52, 51]. Clearly, compared with supervised methods where plain early and late fusion strategies are readily available, there are far fewer options for integration of image features in the unsupervised community.

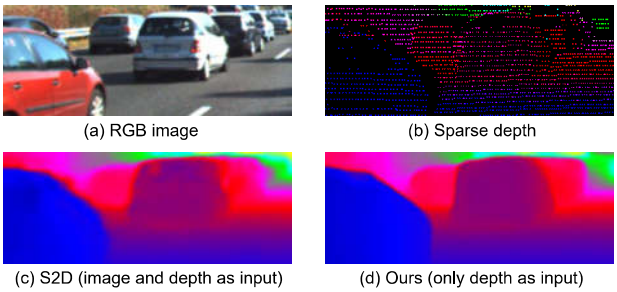


Figure 1. Unsupervised depth completion from sparse depth. Compared with (a) the RGB image, (b) sparse depth presents high structure inconsistency around object boundaries. (c) Existing unsupervised model S2D [32] takes the RGB image as an additional input. (d) Our model only inputs sparse depth. We achieve comparable performance to S2D [32] in producing consistent depth values, especially around object boundaries, even without access to the image at test time.

In this paper, we propose a new approach to integrating image features in unsupervised depth completion. In a nutshell, our method is formulated as an auto-encoder [16, 15], where sparse depth is first transformed into latent features and then recovered into dense depth. The sparse input serves as a supervision signal for the network. Besides the lack of structural cues, a vanilla auto-encoder will not give good performance on depth completion due to its trivial nature, *i.e.*, generating a trivial mapping from input to output as the

[1]Although these methods do not rely on ground truth data and seem to be unsupervised, they are not in the same category with ours because our model involves network learning. In the following, "unsupervised" refers to learning-based depth completion without ground truth.

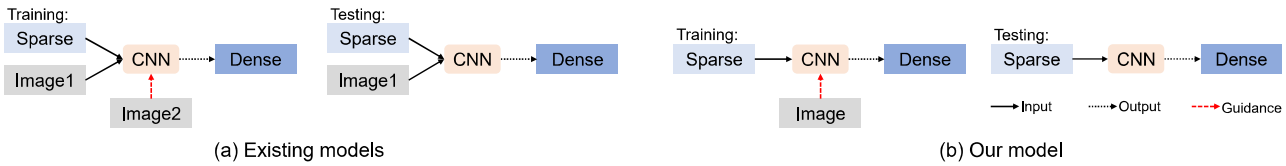


Figure 2. Unsupervised depth completion models. (a) Existing models, *e.g.*, S2D [32] and DDP [55], take the image as an additional input in both training and test phases. A second stereo image constructs the image warping loss, which gives implicit supervision for dense depth. (b) Our model only uses a single image for training. At test time, we recover dense depth from the sparse input only.

input is also used for supervision. To improve performance, we employ an image to guide latent features during training, as illustrated in Fig. 2(b). In addition to providing dense structural cues, the image guidance constrains latent features to reduce the trivial solution. We show that this practice yields a large improvement over the vanilla baseline and allows our method to be competitive on public benchmarks.

We emphasize two distinctive characteristics of our design which make it novel and insightful. First, our method introduces a new setting in unsupervised depth completion, *i.e.*, only using sparse depth as the network input at test time. In comparison, previous unsupervised works adopt the image and sparse depth in both training and test phases, as shown in Fig. 2(a). We demonstrate the feasibility of the new setting with satisfying depth completion accuracy, which benefits the scientific body of literature in this area. Second, we provide insights on the appropriate use of image guidance through various studies, such as the position where image guidance is imposed, impact of feature resolution and the number of channels.

The main points of this paper are summarized below.

1. We propose a new paradigm for unsupervised depth completion that recovers dense depth only from the sparse input at test time. We push this task further beyond existing works that take the image as an additional input and employ a second image for training.

2. Our method is formulated as an auto-encoder and uses the image to directly guide latent features in training. This enables our model to acquire more dense and structural cues, which improve the depth accuracy and maintain structure consistency without the image input.

3. We validate the effectiveness of the proposed image guidance and achieve promising performance on the KITTI Depth Completion Benchmark compared with other unsupervised methods. Our model is also applicable to indoor scenes, *e.g.*, NYUv2.

2. Related Work

Depth completion. Depth completion aims to fill in missing values in the sparse input. Traditional non-learning based methods [24, 40, 3, 12] rely on hand-crafted features such as gradients for completion. More recently, deep learning methods have improved performance substantially. Uhrig *et al*. [45] design sparsity invariant CNNs which generate and update a binary mask (1 for pixels with depth values and 0 for missing ones) to handle sparse data. This binary mask is improved and facilitated with sparsity-invariant properties for more competitive performance [19]. Chodosh *et al*. [8] alternatively utilize compressed sensing to handle sparse data. Eldesokey *et al*. [9, 11] generate a confidence map to measure the reliability of predicted depth values. S2D [32] refines the depth maps iteratively from sparse to dense combined with the image input.

Learning from other features to facilitate depth completion has become a new trend recently. These features normally supply complementary geometric or structural cues to depth and come from various sources, such as RGB images [26, 54], surface normal [35, 53, 1], and point clouds [5, 2]. Some useful techniques are also applied to better aggregate features, *e.g.*, spatial affinity [7, 33, 38], depth coefficients [21], global and local context [46, 6, 7], auxiliary image reconstruction [30], pseudo depth [14], plane/surface extrapolation [25, 20], uncertainty [10, 46], and domain adaptation [29]. However, these always come with complicated networks and a large number of parameters.

Auto-encoders. Auto-encoders aim to generate a compressed feature representation by learning an identity mapping from the input to the output [15, 47]. The input itself is used as a supervision signal for the training process. Recently, deep auto-encoders have been widely employed as an unsupervised learning technique in image denoising [48], super-resolution [37, 57], multi-view learning [50], *etc*.

To improve the discriminative ability of latent features and prevent only learning a trivial mapping from input to output, some constraints are imposed on the latent features. For example, Sundermeyer *et al*. [43] introduce the augmented auto-encoder that controls the encoding of latent features by adding random augmentations to the input. Other useful constraints include graph embeddings [56], non-negativity [17, 44], label consistency [18], and so on. Additionally, latent features can be constrained by being guided/supervised by a certain signal, such as label information [42, 4], pose estimation [27], and feature selection [49]. Our model uses the image to guide latent features, which improves performance by supplying more structural cues to depth and reducing the trivial solution.

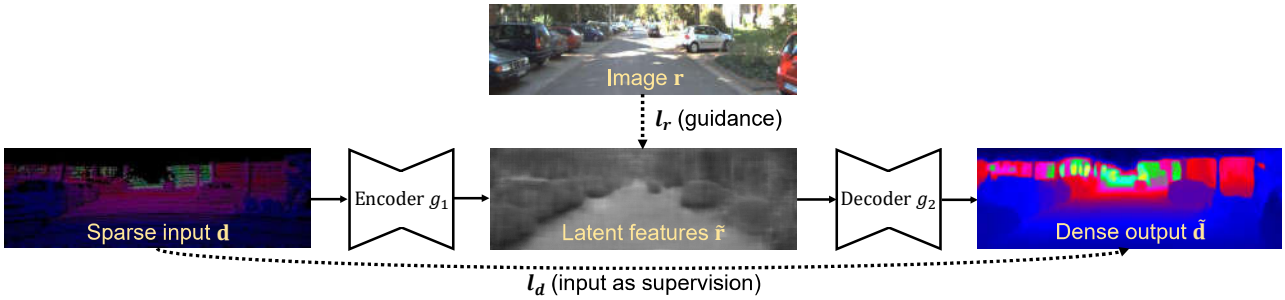


Figure 3. Proposed auto-encoder framework for training unsupervised depth completion. The encoder transforms sparse depth input into latent features, which are then fed into the decoder to produce dense depth. The sparse input itself is used as the supervision signal for depth. By penalizing the difference between the image and latent features during training, the latent features are guided to encode more dense structural cues that are beneficial for producing more accurate and consistent depth values. The latent feature map is obtained from our default model (see Section 5.1) and visualized by normalizing the values into 0-1.

3. Unsupervised Depth Completion Revisited

Unsupervised depth completion models assume there is no dense ground truth or any other manual annotations available. To reduce structure inconsistency resulting from the sparse input $\mathbf{d} \in \mathbb{R}^{H \times W}$ (H and W represent the height and width respectively), existing studies [32, 55, 52, 51] further assume an associated RGB image $\mathbf{r} \in \mathbb{R}^{H \times W \times C}$ is available (C is the number of channels, *e.g.*, 3 for an RGB image and 1 for its grayscale), and take it as an additional input (also see Fig. 2(a)), *i.e.*,

$$\tilde{\mathbf{d}} = f(\mathbf{d}, \mathbf{r}), \tag{1}$$

where $\tilde{\mathbf{d}} \in \mathbb{R}^{H \times W}$ is dense depth output. In this formulation, the sparse input is used as a supervision signal for depth, *i.e.*,

$$\ell_d = \frac{1}{N_1} \left\| \mathbf{M} \odot (\mathbf{d} - \tilde{\mathbf{d}}) \right\|^{\eta}, \tag{2}$$

where $\mathbf{M}$ is a binary mask that indicates validness of input depth (1 for points with depth values and 0 for none), and N_1 is the total number of valid points. η is the norm of the loss, *i.e.*, 1 for ℓ_1 (MAE) and 2 for ℓ_2 (MSE). $\odot$ denotes element-wise multiplication. Additionally, a second image, either from stereo or adjacent frames, is employed to construct the disparity loss [55] or photometric loss [32, 52, 51] (see Fig. 2(a)). This loss is essentially implicit supervision to depth since it does not directly penalize depth reconstruction but the results derived from depth, *i.e.*, the warped image. Without loss of generality, we denote the additional loss as ℓ_c, and thus the entire training loss ℓ_t becomes

$$\ell_t = \ell_d + w_c \cdot \ell_c, \tag{3}$$

where w_c controls the impact of ℓ_c. At test time, the second image is not required, but the image associated with sparse depth is still taken as input.

In addition, [55, 51] have to learn prior information, *e.g.*, dense depth prior or topology prior, with another network on the Virtual KITTI dataset [13]. [32, 52] compute feature correspondences from adjacent images for pose estimation. All of these operations heavily rely on RGB images and other image related information, further indicating the difficulty in integrating image features in this area.

4. Our Method

Section 3 motivates us to reflect on an easier but still effective usage of RGB images. To this end, we formulate our model as an auto-encoder and propose to guide latent features with the image. The general framework is illustrated in Fig. 3. This approach generally has two distinctions: (1) It enables our model to recover dense depth *only* from the sparse input, which is a normal setting in supervised works [30, 45, 11] but has not been well studied in the unsupervised area; (2) It is effective in better keeping structure consistency than using an auto-encoder without image guidance.

4.1. Depth Completion as an Auto-Encoder

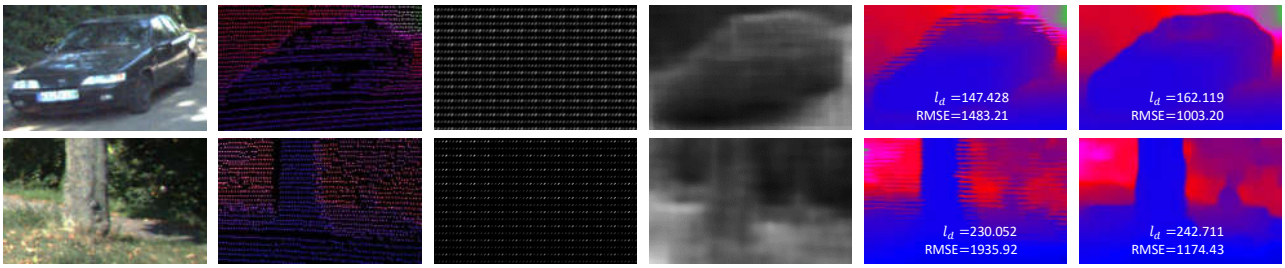
We aim to construct a model g that recovers dense depth only from the sparse input, *i.e.*,

$$\tilde{\mathbf{d}} = g(\mathbf{d}). \tag{4}$$

To achieve this, we regard the dense output as a reconstructed result of the sparse input, and formulate g as an auto-encoder to realize this reconstruction. More specifically, we divide g into an encoder g_1 and a decoder g_2. g_1 transforms the sparse input $\mathbf{d}$ into latent features $\tilde{\mathbf{r}} \in \mathbb{R}^{H_1 \times W_1 \times C_1}$ where H_1 and W_1 represent the height and width, and C_1 is the number of feature channels. g_2 recovers dense depth from $\tilde{\mathbf{r}}$. The entire process is described as:

$$\tilde{\mathbf{d}} = g(\mathbf{d}) \rightarrow \tilde{\mathbf{d}} = g_2(\tilde{\mathbf{r}} = g_1(\mathbf{d})). \tag{5}$$

The model can be trained with the identity mapping loss defined in Eq. 2. We name this model the *vanilla auto-encoder* because it does not incorporate any extra information. Below, we list two major problems with the vanilla auto-encoder.



(a) RGB image (b) Sparse input (c) Vanilla latent features (d) Our latent features (e) Vanilla depth (f) Our depth

Figure 4. Depth completion results based on vanilla auto-encoder and our guided latent features. (a) and (b) are the RGB image (not used as input) and the sparse input. (c) Vanilla latent features directly from sparse depth are also highly sparse, and they cannot indicate any clear or useful structural information. (d) Our image-guided latent features, by contrast, are able to acquire more dense and structural cues, *e.g.*, the general shapes of the car and tree are clearer than (c). (e) Dense depth from the vanilla auto-encoder fails to complete object boundaries properly. It has a smaller difference ℓ_d with the input, but larger errors, *e.g.*, RMSE, compared with the ground truth. (f) Our depth with guided latent features presents more visually consistent boundaries. It also produces more correct depth values, indicating the reduced impact of the trivial solution as ℓ_d is slightly larger, but the RMSE is much smaller. The latent feature map is obtained from our default model (see Section 5.1) and visualized by normalizing the values into 0-1.

Insufficient structural cues. Without additional guidance, *e.g.*, the image, both the sparse input and its latent features cannot provide sufficient structural cues for accurate depth completion, particularly around object boundaries. For example, in Fig. 4(c), the latent features of the sparse input are still highly sparse, and we can hardly find any clear and useful structural information of the car and tree from them. The completed results based on these features present inconsistent depth values around boundaries (see Fig. 4(e)). Hence, it is difficult to recover consistent and accurate dense depth only from the sparse input.

Trivial solution. g takes the sparse input $\mathbf{d}$ as both input and supervision, which may produce a trivial solution that $\tilde{\mathbf{d}}$ is infinitely close to $\mathbf{d}$ in valid positions that contain input values. The accuracy of other missing values to be completed is largely sacrificed. As shown in Fig. 4(e) and (f), even though the difference between the output and the sparse input is smaller with the vanilla model, the errors, *i.e.*, RMSE, are larger. This can also be reflected by visual results, where stripe artifacts with similar patterns to horizontally scanned LiDAR points, exist around object boundaries. The negative impact of the trivial solution should be reduced.

4.2. Image Guidance to Latent Features

To deal with above issues, we propose to use the image to guide latent features in the training process (see Fig. 3). It aims to regularize latent features to obtain more structural cues from the image and prevent the trivial solution. We define a function ϕ that converts the image $\mathbf{r} \in \mathbb{R}^{H \times W \times C}$ into the image feature representation $\phi(\mathbf{r}) \in \mathbb{R}^{H_1 \times W_1 \times C_1}$ that shares identical feature dimensions with latent features $\tilde{\mathbf{r}}$. ϕ is either (1) a self mapping, *i.e.*, $\phi(\mathbf{r}) = \mathbf{r}$, or (2) a CNN to extract convolutional features. The guidance works by penalizing the difference between the two features, *i.e.*,

$$\ell_r = \frac{1}{H_1 W_1 C_1} \left\| \phi(\mathbf{r}) - \tilde{\mathbf{r}} \right\|^{\gamma}, \tag{6}$$

where γ determines the norm of the loss. Combined with the sparse depth loss defined in Eq. 2, the total training loss of the proposed model is

$$\ell_{total} = \ell_d + w_r \cdot \ell_r, \tag{7}$$

where w_r weighs the impact of the image guidance loss ℓ_r. The encoder and decoder share the same network architecture, *i.e.*, they follow the standard U-shaped structure. More details on the network configuration can be found in the supplementary material.

4.3. Discussion

Why can dense depth be directly constructed from sparse input only? For a specific position in sparse depth, convolving with a squared kernel is like performing a weighted sum within the local region. If that position has no depth value, it will be updated based on nearby points with values. This is the underlying reason that dense depth can be directly constructed from the sparse input. Supervised by valid points in the input, the weights are learnable.

The role of image guidance. Image guidance enables latent features to better acquire dense and structural cues that can facilitate depth completion. In Fig. 4(d), the guided latent features of the car and tree reveal their general shapes more clearly than vanilla (unguided) ones that only have sparse representations. As illustrated in Fig. 4(f), both examples have a larger ℓ_d but lower RMSE than vanilla results, also indicating the reduced impact of the trivial solution.

The proposed image guidance is inspired by [30], a fully-supervised model that acquires image features by reconstructing the image from sparse depth. The similarity with ours is that both works add the image loss as part of the training loss. However, the underlying insights of such image guidance are different. In terms of the network architecture, our method does not have an image decoder separate from

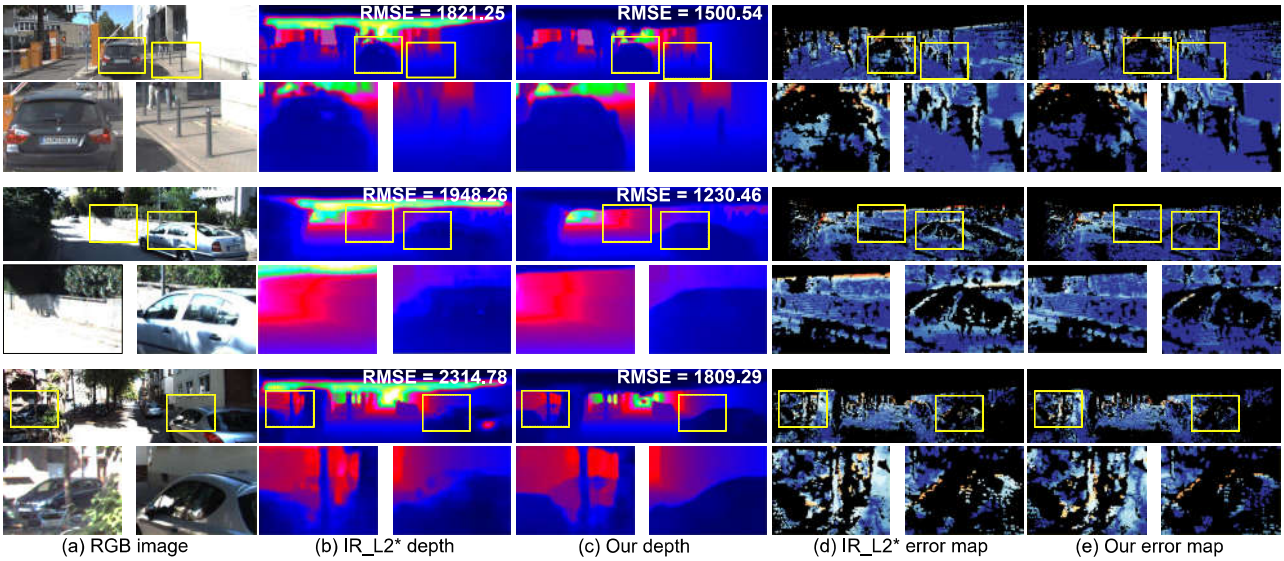


Figure 5. Qualitative comparison with IR_L2* [30] retrained with the unsupervised setting on the KITTI test set. Our RMSE in three examples is better than IR_L2* [30], and our results present smaller errors, especially around object boundaries, according to close-up error maps. It indicates the effectiveness of the proposed explicit image guidance to latent depth features.

	Method	#Param.	RMSE ↓	MAE ↓	iRMSE ↓	iMAE ↓
Unsupervised (only sparse input)	IR_L2*	11.63M	1943.28	541.36	17.88	7.76
	Ours	2.29M	1451.67	429.74	4.89	1.78
Unsupervised (sparse & RGB inputs)	S2D	18.8M	1299.85	350.32	4.07	1.57
	DDP	27.8M	1263.19	343.46	3.58	1.32
	VOICED	9.7M	1169.97	299.41	3.56	1.20
	ScaffFusion	7.8M	1121.89	282.86	3.32	1.17

Table 1. Quantitative comparison with unsupervised methods on the KITTI test set. The methods we compare include IR_L2* [30] (this method is retrained with the unsupervised setting, *i.e.*, replacing dense ground truth with input sparse depth), S2D [32], DDP [55], VOICED [52], and ScaffFusion [51]. These results are obtained from the benchmark, and no ground truth is available. ↓ means smaller is better.

the main branch, so it does not aim to reconstruct the image. Functionally, our image guidance is directly imposed on latent depth features by penalizing their difference (regarded as *explicit guidance*), and dense depth has to be recovered from the refined features. By contrast, [30] implicitly generates image-related features by reconstructing the image as an output. We will justify the effectiveness of our explicit image guidance in the unsupervised setting through experiments.

Relationship with existing unsupervised models. Our formulation for training the model in Eq. 7 is consistent with the general form of the unsupervised framework defined in Eq. 3. The image guidance loss ℓ_r, similar to ℓ_c in Eq. 3, is an extra loss that facilitates network training. However, it is essentially different from other works [32, 55, 52, 51] in that (1) it focuses on enhancing intermediate latent features, and (2) it does not require a second image for training.

Inference. Learning the proposed depth completion autoencoder only requires the image during training. At test time, our model only takes sparse depth as input (see Fig. 2(b)), *i.e.*,

$$\tilde{\mathbf{d}} = g(\mathbf{d}; \theta_g^*), \tag{8}$$

where θ_g^* denotes the parameters of the optimal model.

5. Experiments

In this section, we demonstrate the effectiveness of our method through both quantitative and qualitative results.

5.1. Experimental Details

Dataset. We report depth completion results on the KITTI Depth Completion Benchmark [45]. The KITTI depth maps are acquired by reprojecting LiDAR points taken over a short time window onto an image, and around 5% of the pixels have depth values. When counting the sparse depth maps, there are 85,898 training, 1,000 validation, and 1,000 test images in total. Ground truth depth maps are generated by accumulating LiDAR points from adjacent frames using semi-global matching, with outliers manually removed [45]. Test set performance is evaluated on the online benchmark server with no ground truth available to the public.

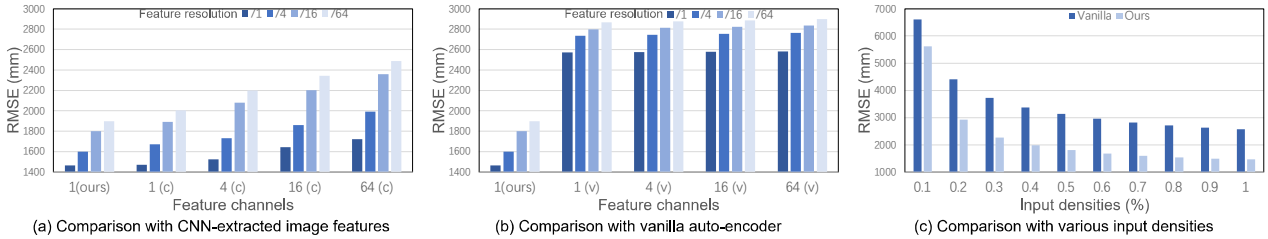


Figure 6. Model analysis. (a) Impact of the resolution and number of channels of latent features. "c" means that we use CNNs to extract the image features. (b) Comparison with the vanilla auto-encoder with different feature resolutions and channels. "v" represents the vanilla auto-encoder. (c) Robustness to input densities. Here the vanilla auto-encoder share the same latent feature resolution and channel with our default image-guided model.

Evaluation metrics. Following the benchmark [45], we use four quantitative evaluation metrics: (1) root mean square error (RMSE in mm), (2) mean absolute error (MAE in mm), (3) RMSE of inverse depth (iRMSE in 1/km), and (4) MAE of inverse depth (iMAE in 1/km). Among them, RMSE is used to rank approaches on the benchmark.

Training procedure. We implement our network with PyTorch [34], and train and test the model on one NVIDIA Titan X GPU. All the training data have a resolution of 352×1216. The model is trained with the Adam optimizer [23], where the initial learning rate is set as 0.001. In our default model, $\eta = 1$, $\gamma = 1$, $w_r = 0.1$. Further, the latent features $\tilde{\mathbf{r}} \in \mathbb{R}^{352\times1216\times1}$ share the same spatial resolution, *i.e.*, height and width, with the input, and only have one feature channel that is directly guided by the gray-scale image without any convolutional layers to extract image features, *i.e.*, $\phi(\mathbf{r}) = \mathbf{r} \in \mathbb{R}^{352\times1216\times1}$. The default vanilla auto-encoder share the same latent feature resolution and channel with our default model. All the other details of parameter settings can be found in the supplementary material.

5.2. Comparison with Existing Methods

We compare four published unsupervised works, S2D [32], and DDP [55], VOICED [52], and ScaffFusion [51]. Note that although IR_L2 [30] is not specially designed for unsupervised depth completion, its usage of the image is similar to ours, *i.e.*, adding it to the training loss. For a fair comparison, we retrain IR_L2 [30] by replacing dense ground truth with sparse depth. We rename it as IR_L2*.

IR_L2*. We report quantitative results in Table 1. Our model significantly outperforms IR_L2* [30], *i.e.*, surpassing RMSE by 491.61 (25.3%), MAE by 111.62 (20.6%), iRMSE by 12.99 (72.65%), and iMAE by 5.98 (77.1%). Further, we have around 5× fewer parameters than IR_L2* [30]. Qualitative results in Fig. 5 also indicate the superiority of our model in key regions such as object boundaries.

The primary reason for our superior performance is that the proposed image guidance gives direct and explicit refinement to depth features (a "brute-force" refinement). By contrast, IR_L2* implicitly learns image-related features by reconstructing the image from sparse depth and then transfers them to the depth completion encoder. This is less helpful when dense depth ground truth is unavailable, because it is more difficult for depth features to coincide with image features with such limited depth points for supervision.

S2D, DDP, VOICED, and ScaffFusion. The quantitative results on the KITTI test set are reported in Table 1. Naturally, our method does not beat the four works quantitatively due to the input difference and less additional information used during training. Even so, we still achieve competitive performance in some qualitative examples, which are displayed in the supplementary material.

5.3. Analysis

Impact of the resolution and number of channels of latent features. For clarity, the feature resolution refers to the spatial dimension, *i.e.*, height and width, and channels represent the number of feature maps. We first investigate their impact to our image-guided model. For the self-mapping, we set the latent channel number to 1, and then use the one-channel gray-scale image to directly guide latent features. For the CNN mapping, we apply a 3-layer convolutional network with 3×3 kernels to extract image features from the gray-scale image. From Fig. 6(a), we observe that using CNNs to extract image features does not bring significant performance gain, *i.e.*, using the original image to directly guide latent features yields the best RMSE in all feature resolutions. Moreover, adding extra channels with CNNs reduces performance. We attribute the performance degradation after using more parameters to the *over-complete auto-encoder* [48], *i.e.*, the model tends to simply copy the input to the output without learning useful features. This problem is caused by having excessive parameters in the hidden layer, *e.g.*, using too many channels or a very complex network. In our system, the sparse input only has one channel, so we design the latent feature to have one channel and the network to be light-weight.

For feature resolution, we find that reducing it leads to larger errors. This is because the spatial correspondence at each position between the input, image, and output cannot be well preserved with the reduced resolution. We show visual examples in the supplmentary material. These results

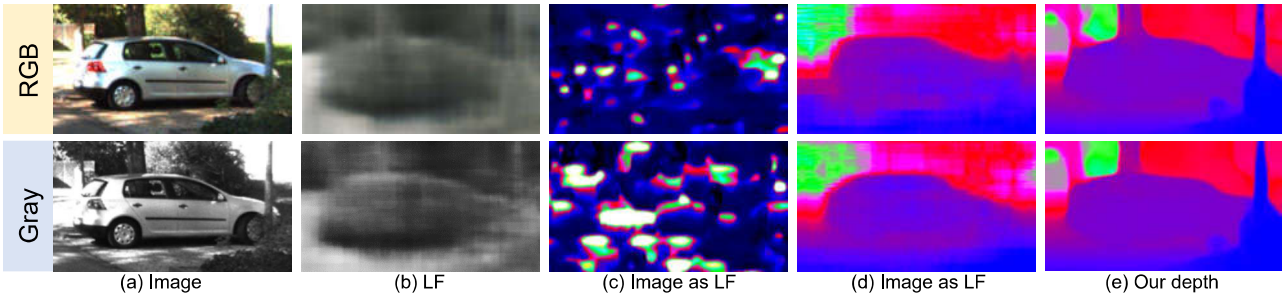


Figure 7. Qualitative results of using RGB and gray images to guide or replace latent features. There is no significant difference between using the RGB or gray image to guide latent features. Replacing latent features with the image, either retraining or not retraining the model, cannot produce better results than ours. "LF" represents latent features, and they are visualized by normalizing the values into 0-1.

	Method	RMSE ↓	MAE ↓
Unsupervised (only sparse input)	Vanilla	2572.71	696.53
	Vanilla▲	1149.93	371.12
	Vanilla★	2857.28	761.62
	Ours	1464.69	431.23
	Ours▲	1077.46	359.46
	Ours★	1542.16	445.60
	EG	1985.38	583.01
	DG	2496.80	682.19
	OG	2723.14	711.77

Table 2. Quantitative comparison with the vanilla auto-encoder and different positions of image guidance on the KITTI validation set. ▲ means the model is evaluated on valid pixels in the input that have ground truth. ★ is the evaluation on remaining pixels that do not have input values but have ground truth. "EG", "DG", and "OG" mean image guidance is placed to the encoder, decoder, and output respectively. ↓ means smaller is better.

validate our choice in default vanilla and image-guided models, *i.e.*, one-channel latent feature with full resolution. In the following, all the experiments are performed on default models unless otherwise specified.

Effectiveness of image guidance over the vanilla auto-encoder. To validate the effectiveness of image guidance, we compare the vanilla auto-encoder and our image-guided model. In Fig. 6(b), our model outperforms the vanilla one by a large margin with various settings of feature resolutions and channels. Further, combining Fig. 6(a) and (b), the overall performance after using image guidance is significantly better than the vanilla model in all the cases. For quantitative results in Table 2, our model surpasses its RMSE by 1108.02 (43.1%) and MAE by 265.3 (38.1%). We further separately evaluate valid pixels in the input and remaining pixels that do not have input values in the table. The performance on valid pixels is similar because they are input values commonly existing in two models. However, our model significantly outperforms the vanilla one on remaining pixels. This large improvement is owing to image guidance, and indicates that our method can generalize better to unseen pixels.

Reasons for intermediate guidance. To verify intermediate guidance to latent features, we place this guidance to the encoder (EG), the decoder (DG), and output (OG) respectively, and then retrain the model. From Table 2, employing image guidance to the encoder produces slightly worse results, which is because depth features from the sparse input have not been sufficiently encoded. By contrast, the performance after moving the guidance to the decoder is significantly degraded. The underlying reason is the decoder's task is to recover dense depth from latent features. Requiring an additional task of the decoder detracts from the depth completion task. An extreme case is after we place image guidance at the output layer, the results become even worse than the vanilla model (the decoder has to both recover dense depth and reconstruct the image, which is difficult to work well). In conclusion, applying image guidance to intermediate latent features yields the best results, where the original depth features have been well encoded and refined, and the decoder can focus on depth completion.

RGB *vs.* gray guidance. We can use either the RGB image or its grayscale to guide latent features. They do not differ too much in terms of the final performance because the image content in two color spaces is similar, *e.g.*, important structures contained in RGB are also mostly visible in gray. From Fig. 7(b) and (e), we do not see obvious difference between two types of latent features and dense output, except that latent features guided by the gray image present brighter appearance. According to quantitative results in Table 3, we find that using the gray image for guidance yields slightly better results. This is because it is harder to penalize RGB and latent features as it involves three channels. Also, more feature channels make the model easier to be affected by the over-complete issue (see above). In fact, in Fig. 7(b), we show that the learned 3-channel latent feature (visualized like an RGB image) does not present obvious colors, *i.e.*, it still looks gray. It suggests that the network does not rely on specific colors for completion. Structure informa-

	Method	RMSE ↓	MAE ↓
Guiding LF	RGB	1485.85	439.76
	Gray	1464.69	431.23
Replacing LF (w/o retraining)	RGB	24499.30	9209.43
	Gray	16107.76	8867.21
Replacing LF (with retraining)	RGB	4615.71	2003.55
	Gray	5134.19	2321.09

Table 3. Quantitative results of using RGB and gray images to guide or replace latent features on the KITTI validation set. ↓ means smaller is better. There is not significant difference between using the RGB or gray image to guide latent features. Replacing latent features with the image, either with or without retraining, produces poor results. "LF" represents latent features.

tion indicated by intensity difference is more important. A similar phenomenon on better results with the gray image is observed in [31, 30]. Moreover, in terms of practical use, the gray image occupies less storage than RGB. Hence, by default, we use the gray image to guide latent features.

Replacing latent features with the image. The image guidance loss defined in Eq. 6 enforces the similarity between latent features and the image. A natural question is, what will the performance be if we replace latent features with the image, *i.e.*, the image guidance loss is equal to zero?

The first experiment is, given our trained model with default settings, we replace latent features with the image directly at test time. As shown in Fig. 7(c), the decoder cannot recover any correct depth, which is also reflected by the extremely poor quantitative results in Table 3. The underlying reason is that the trained parameters are fixed, and the direct replacement destroys the learned information in latent layers. The second experiment is that we replace latent features with the image and retrain the entire model. This makes more sense as it actively adjusts parameters. In that case, the encoder is blocked and the network comes to use the decoder to recover dense depth directly from the image, *i.e.*, depth estimation from a single image supervised by the sparse input. We can observe in Fig. 7(d) as well as Table 3 that this approach produces better results than above model without retraining. However, the performance is still less competitive than ours. Visual results indicate that the depth of some important details, *e.g.*, trees and car boundaries, cannot be properly recovered.

Based on these results, we find that replacing latent features with the image is less effective for depth completion. Specifically, latent features guided by the image and the image itself are two different concepts. Latent features are a type of feature representation of sparse depth. Guided by the image, they are embedded with more consistent structural cues, but are still conditioned on sparse depth rather than the image. By contrast, the image is another modality inherently different from depth. It cannot be regarded as a latent representation of sparse depth, so recovering depth directly from the image is less accurate. This is also observed in [28, 36].

	Method	RMSE ↓	REL ↓
Unsupervised (only sparse input)	Vanilla	0.449	0.081
	IR_L2* [30]	0.358	0.062
	Ours	0.315	0.053
Hand-crafted (sparse & RGB inputs)	TGV [12]	0.635	0.123
	Bilateral [41]	0.479	0.084

Table 4. Quantitative comparison on the NYUv2 test set. ↓ means smaller is better. Our model outperforms the vanilla auto-encoder, IR_L2*[30], and hand-crafted methods. It indicates that our method has good applicability to other dataset.

Robustness to input densities. We also analyze the robustness to different input densities. The valid points with depth values in the original sparse input account for around 5% of the entire depth map. We further reduce the input sparsity by randomly retaining points with ratios from 90% to 10%, similar to [39, 53, 7, 28]. Our results in Fig. 6(c) demonstrate the good generalization ability of our model to different densities (measured by RMSE). With an increased density, depth completion performance is gradually enhanced because more input data is provided. Moreover, our model performs consistently better in all cases than the vanilla auto-encoder, which further indicates its effectiveness. We also reduce the LiDAR scanning lines as an alternative way to downsample the input, which is illustrated in the supplementary material.

Application to indoor scenarios. Our model can also be applied to indoor scenes, *e.g.*, NYUv2 [41]. Each sparse input for training has 500 randomly selected depth values, the same as [30, 35]. We evaluate the proposed model on the official labelled test set that contains 654 samples. In Table 4, we report RMSE and REL (mean absolute relative error). Our model outperforms the vanilla auto-encoder, IR_L2* [30], and hand-crafted methods.

6. Conclusion

In this paper, we propose a new unsupervised depth completion model. Formulated as an auto-encoder, our model only takes sparse depth as input, which is essentially different from existing unsupervised works that use the RGB image as an additional input in both training and test phases. To reduce structure inconsistency, we propose to employ the image to guide latent features in the training process. This approach enables the acquisition of more dense and structural features beneficial for producing more consistent and accurate depth values. We validate its effectiveness through extensive experiments on the KITTI Depth Completion Benchmark. The proposed method also has good applicability to indoor scenes, *e.g.*, NYUv2. Our future work will focus on leveraging other information, *e.g.*, 3D point clouds, surface normal, to enhance latent features.

References

[1] Pei An, Yingshuo Gao, Wenxing Fu, Jie Ma, Bin Fang, and Kun Yu. Lambertian model based normal guided depth completion for lidar-camera system. *IEEE GRSL*, pages 1–4, 2021.

[2] Lin Bai, Yiming Zhao, Mahdi Elhousni, and Xinming Huang. Depthnet: Real-time lidar point cloud depth completion for autonomous vehicles. *IEEE Access*, 8:227825–227833, 2020.

[3] Jonathan T Barron and Ben Poole. The fast bilateral solver. In *European Conference on Computer Vision*, pages 617–632. Springer, 2016.

[4] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, 2007.

[5] Yun Chen, Bin Yang, Ming Liang, and Raquel Urtasun. Learning joint 2d-3d representations for depth completion. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[6] Xinjing Cheng, Peng Wang, Guan Chenye, and Ruigang Yang. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. *Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*, 2020.

[7] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–119, 2018.

[8] Nathaniel Chodosh, Chaoyang Wang, and Simon Lucey. Deep convolutional compressed sensing for lidar depth completion. In *Asian Conference on Computer Vision*, pages 499–513. Springer, 2018.

[9] Abdelrahman Eldesokey. Propagating confidences through cnns for sparse data regression. In *The British Machine Vision Conference (BMVC)*, 2018.

[10] Abdelrahman Eldesokey, Michael Felsberg, Karl Holmquist, and Michael Persson. Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[11] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through cnns for guided sparse depth regression. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[12] David Ferstl, Christian Reinbacher, Rene Ranftl, Matthias Rüther, and Horst Bischof. Image guided depth upsampling using anisotropic total generalized variation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 993–1000, 2013.

[13] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016.

[14] J. Gu, Z. Xiang, Y. Ye, and L. Wang. Denselidar: A real-time pseudo dense depth guided depth completion network. *IEEE Robotics and Automation Letters*, 6(2):1808–1815, 2021.

[15] Hinton, G., E., Salakhutdinov, R., and R. Reducing the dimensionality of data with neural networks. *Science*, 2006.

[16] Geoffrey E. Hinton and Richard S. Zemel. Autoencoders, minimum description length and helmholtz free energy. *Advances in neural information processing systems*, 6, 1993.

[17] Ehsan Hosseini-Asl, Jacek M. Zurada, and Olfa Nasraoui. Deep learning of part-based representation of data using sparse autoencoders with nonnegativity constraints. *IEEE Transactions on Neural Networks & Learning Systems*, 27(12):2486–2498, 2016.

[18] Cong Hu, Xiao-Jun Wu, and Zhen-Qiu Shu. Discriminative feature learning via sparse autoencoders with label consistency constraints. *Neural Processing Letters*, 2018.

[19] Zixuan Huang, Junming Fan, Shenggan Cheng, Shuai Yi, Xiaogang Wang, and Hongsheng Li. Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion. *IEEE Transactions on Image Processing*, 29:3429–3441, 2019.

[20] Saif Imran, Xiaoming Liu, and Daniel Morris. Depth completion with twin surface extrapolation at occlusion boundaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2583–2592, 2021.

[21] Saif Imran, Yunfei Long, Xiaoming Liu, and Daniel Morris. Depth coefficients for depth completion. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12438–12447. IEEE, 2019.

[22] Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. In *2018 International Conference on 3D Vision (3DV)*, pages 52–60. IEEE, 2018.

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] Johannes Kopf, Michael F Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. In *ACM Transactions on Graphics (ToG)*, volume 26, page 96. ACM, 2007.

[25] Byeong-Uk Lee, Kyunghyun Lee, and In So Kweon. Depth completion using plane-residual representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13916–13925, 2021.

[26] Ang Li, Zejian Yuan, Yonggen Ling, Wanchao Chi, Chong Zhang, et al. A multi-scale guided cascade hourglass network for depth completion. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 32–40, 2020.

[27] Zhigang Li and Xiangyang Ji. Pose-guided auto-encoder and feature-based refinement for 6-dof object pose regression. In *IEEE International Conference on Robotics and Automation (ICRA), 2020*, 2020.

[28] Lina Liu, Yiyi Liao, Yue Wang, Andreas Geiger, and Yong Liu. Learning steering kernels for guided depth completion. *IEEE Transactions on Image Processing*, 30:2850–2861, 2021.

[29] Adrian Lopez-Rodriguez, Benjamin Busam, and Krystian Mikolajczyk. Project to adapt: Domain adaptation for depth completion from noisy and sparse sensor data. In *Asian Conference on Computer Vision (ACCV)*, 2020.

[30] Kaiyue Lu, Nick Barnes, Saeed Anwar, and Liang Zheng. From depth what can you see? depth completion via auxiliary image reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[31] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. *arXiv preprint arXiv:1807.00275*, 2018.

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

[33] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *European Conference on Computer Vision (ECCV)*, 2020.

[34] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

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

[36] Chao Qu, Ty Nguyen, and Camillo Taylor. Depth completion via deep basis fitting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 71–80, 2020.

[37] Chen Rong, Yanyun Qu, Cuihua Li, Kun Zeng, and Ce Li. Single-image super-resolution via joint statistic models-guided deep auto-encoder network. *Neural Computing & Applications*, 32(2):1–12, 2018.

[38] René Schuster, Oliver Wasenmüller, Christian Unger, and Didier Stricker. Ssgp: Sparse spatial guided propagation for robust and generic interpolation. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[39] René Schuster, Oliver Wasenmüller, Christian Unger, and Didier Stricker. Ssgp: Sparse spatial guided propagation for robust and generic interpolation. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021.

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

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

[42] Jasper Snoek, Ryan P. Adams, and Hugo Larochelle. Nonparametric guidance of autoencoder representations using label information. *Journal of Machine Learning Research*, 13(1):2567–2588, 2012.

[43] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 699–715, 2018.

[44] Yueyang Teng, Yichao Liu, Jinliang Yang, Chen Li, Shouliang Qi, Yan Kang, Fenglei Fan, and Ge Wang. Graph regularized sparse autoencoders with nonnegativity constraints. *Neural Processing Letters*, 50(1):247–262, 2019.

[45] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 International Conference on 3D Vision (3DV)*, pages 11–20. IEEE, 2017.

[46] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. In *International Conference on Machine Vision Applications (MVA)*, 2019.

[47] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, 2008.

[48] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(12):3371–3408, 2010.

[49] Shuyang Wang and Zhengming Ding. Feature selection guided auto-encoder. In *AAAI*, 2017.

[50] Shuyang Wang, Zhengming Ding, and Yun Fu. Coupled marginalized auto-encoders for cross-domain multi-view learning. In *IJCAI*, pages 2125–2131, 2016.

[51] Alex Wong, Safa Cicek, and Stefano Soatto. Learning topology from synthetic data for unsupervised depth completion. *IEEE Robotics and Automation Letters*, 6(2):1495–1502, 2021.

[52] Alex Wong, Xiaohan Fei, Stephanie Tsuei, and Stefano Soatto. Unsupervised depth completion from visual inertial odometry. *IEEE Robotics and Automation Letters*, 5(2):1899–1906, 2020.

[53] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Junhu Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[54] L. Yan, K. Liu, and E. Belyaev. Revisiting sparsity invariant convolution: A network for image guided depth completion. *IEEE Access*, pages 1–1, 2020.

[55] Yanchao Yang, Alex Wong, and Stefano Soatto. Dense depth posterior (ddp) from single image and sparse range. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[56] Wenchao Yu, Guangxiang Zeng, Ping Luo, Fuzhen Zhuang, Qing He, and Zhongzhi Shi. Embedding with autoencoder regularization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 208–223. Springer, 2013.

[57] Kun Zeng, Jun Yu, Ruxin Wang, Cuihua Li, and Dacheng Tao. Coupled deep autoencoder for single image super-resolution. *IEEE Transactions on Cybernetics*, pages 27–37, 2015.