Dynamic Spatial Propagation Network for Depth Completion

Yuankai Lin\textsuperscript{1}, Tao Cheng\textsuperscript{2,}, Qi Zhong\textsuperscript{3}, Wending Zhou\textsuperscript{4} and Hua Yang\textsuperscript{1}

\textsuperscript{1}School of Mechanical Science and Engineering, Huazhong University of Science and Technology, China
\textsuperscript{2}College of Urban Transportation and Logistics, Shenzhen Technology University, China
\textsuperscript{3}Division of Logistics and Transportation, Tsinghua University, China
\textsuperscript{4}School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen), China
\{linyuankai,huayang\}@hust.edu.cn, chengtao@sztu.edu.cn, zhongq20@mails.tsinghua.edu.cn, 220019048@link.cuhk.edu.cn

Abstract

Image-guided depth completion aims to generate dense depth maps with sparse depth measurements and corresponding RGB images. Currently, spatial propagation networks (SPNs) are the most popular affinity-based methods in depth completion, but they still suffer from the representation limitation of the fixed affinity and the over smoothing during iterations. Our solution is to estimate independent affinity matrices in each SPN iteration, but it is over-parameterized and heavy calculation. This paper introduces an efficient model that learns the affinity among neighboring pixels with an attention-based, dynamic approach. Specifically, the Dynamic Spatial Propagation Network (DySPN) we proposed makes use of a non-linear propagation model (NLPM). It decouples the neighborhood into parts regarding to different distances and recursively generates independent attention maps to refine these parts into adaptive affinity matrices. Furthermore, we adopt a diffusion suppression (DS) operation so that the model converges at an early stage to prevent over-smoothing of dense depth. Finally, in order to decrease the computational cost required, we also introduce three variations that reduce the amount of neighbors and attentions needed while still retaining similar accuracy. In practice, our method requires less iteration to match the performance of other SPNs and yields better results overall. DySPN outperforms other state-of-the-art (SoTA) methods on KITTI Depth Completion (DC) evaluation by the time of submission and is able to yield SoTA performance in NYU Depth v2 dataset as well.

Introduction

Dense depth estimation is essential for 3D vision tasks, e.g., 3D object detection (Ma et al. 2020b) and tracking (Hu et al. 2019), 6D object pose estimation (Wang et al. 2019), simultaneously localization and mapping (SLAM) (Zhang and Singh 2015), and structure-from-motion (SfM) (Zhu et al. 2018), which are widely used in autonomous driving and robot perception. However, the output of depth sensors such as LiDAR and ToF cameras is sparse, resulting in a large number of empty data regions, which need further processing to fulfill or simply ignore these missing regions.

\textsuperscript{*}Corresponding author

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
the representation capability of SPNs. Taking CSPN (Cheng, Wang, and Yang 2018) as an example, as illustrated in Fig. 1, when the kernel size is 7, most of the affinity weights are distributed in the distance, which means that the affinity matrix cannot clearly represent the relationship between pixels and neighbors.

To solve the problems mentioned above, inspired by Dynamic filters (Jia et al. 2016), we proposed a new non-linear propagation design, named Dynamic Spatial Propagation Network (DySPN), which can learn an adaptive affinity matrix. To implement this design, different attention levels are assigned to neighbors of different distances when refining each pixel in the depth map recursively. More specifically, at the beginning of the propagation, far neighbors provide long-range information to fill the hole of the initial depth map and smooth it quickly. As the depth map becomes denser, near neighbors are paid more attention to edge-preserving. Our attention mechanism adjusts the affinity matrix according to each propagation stage, making the process more accurate and efficient. Besides, we adopt a diffusion suppression operation, which estimates the affinities of refining depths in every step to preserve good parts of the depths and avoid over-smoothing. The affinity of refining depth is easy to participate in the propagation process with nearly no additional computational cost. Although the computational complexity of our DySPN is similar to CSPN, we try to use a larger receptive field and fewer neighbors to achieve a better result. Thus, we introduce three variations that reduce the amount of distant pixels needed dramatically while still retaining similar accuracy. Compared with other SPNs, the iteration steps of our proposed method is only half, which is 6 in this paper.

In summary, our contribution lies in three aspects:

• We propose a non-linear propagation model (NLPM) named DySPN. This model adjusts affinity weights dynamically which is a new prospect to improve SPN. We further achieve it by neighborhood decoupling and spatial-sequential attention.
• A diffusion suppression (DS) operation is proposed, which can adaptive terminate the propagation and reduce over-smoothing.
• Three implementations of the DySPN model are proposed, which can effectively reduce computational complexities and improve performance.

Related Work

Image-guided depth completion Early depth completion methods (Hawe, Kleinsteuber, and Diepold 2011) (Yang, Kim, and Park 2012) (Chiu, Leou, and Hsiao 2014) only rely on sparse measurements to reconstruct or fill holes. Image-guided depth completion methods (Yang et al. 2007) (Park et al. 2011) (Ma and Karaman 2018) usually achieve better results because the additional RGB images contain a large amount of surface, edge, or semantic information.

Later works conduct the guidance or fusion strategies of these RGB image information. For example, DeepLiDAR (Qiu et al. 2019) is a network composed of a color pathway and a surface normal pathway. ACMNet (Zhao et al. 2021) adopts the graph propagation to capture the observed contextual information in the encoder stage. Inspired by the guided image filtering, GuideNet (Yang et al. 2021) proposes an image guidance convolution module for multi-modal feature fusion. FCFR-NET (Liu et al. 2021) designs an energy-based fusion operation to fuse the features of color and depth information. PENet (Hu et al. 2021) uses addition operation to merge color and depth features of different modalities at multiple stages. Besides, KBNet (Wong and Soatto 2021) transfers spatial positional encoding across different branches of the encoder by a calibrated backprojection module.

These methods usually use two independent branches to extract the features of RGB images and sparse depth maps and fuse them at different scales.

Spatial propagation networks SPNs are depth refinement technologies that differ from the aforementioned two-branch architectures. They are usually connected by a simple Unet (Ronneberger, P.Fischer, and Brox 2015) network. Among them, SPN (Liu et al. 2017) is the first proposed method to use a deep CNN to learn an affinity matrix that effectively models any pairwise similarity matrix. SPN builds a row/column linear propagation model whose spatially varying transformation matrix constitutes the affinity matrix. It can be easily applied to many high-level vision tasks, including depth completion and semantic segmentation. Later, CSPN (Cheng, Wang, and Yang 2018) further improves the linear propagation model and adopts a recursive convolution operation to be more efficient. CSPN++ (Cheng et al. 2020) merges the outputs of three independent CSPN modules so that its propagation learns adaptive convolutional kernel sizes and the number of iterations. NLSPN (Park et al. 2020) predicts the relevant non-local neighbors to avoid the mixed-depth problem caused by irrelevant local neighbors. DSPN (Xu, Yin, and Yao 2020) utilizes their deformable SPN that adaptively adjusts receptive fields at each pixel. ODE-CNN (Cheng et al. 2020b) adopts an inverse gnomonic projection CSPN and a deformable CSPN, which better recovers the structural details of Omnidirectional depth maps. PENet (Hu et al. 2021) implement a dilated CSPN++ to enlarge the propagation neighborhoods.

However, these methods are constrained by the bottleneck of the linear propagation model, as the affinity matrix is fixed during propagation. Fixed affinity weights make it difficult for propagation to take into account both long-range dependencies and local information, resulting in more SPN iterations.

Dynamic filters Dynamic filters (Jia et al. 2016) (Su et al. 2019) could dynamically modify or predict filter weights based on the input features, and has shown a strong connection to the attention mechanism (Xu and Saenko 2016). CondConv (Yang et al. 2019) parameterizes several convolutional kernels as a linear combination to mix the experts voted. WeightNet (Ma et al. 2020a) directly generates the weights by a grouped fully-connected layer which is different from common practice that applies operations on vectors in feature space. DyNet (Zhao et al. 2020) presents a dynamic convolution kernel generation method whose coef-
efficient prediction module is trainable and designed to predict the coefficients of fixed convolution kernels. DynamicConv (Chen et al. [2020]) utilizes an Attention mechanism to aggregate multiple convolution kernels to increase the model capability. Furthermore, DDF (Zhou et al. [2021]) simultaneously applies spatial and channel-wise attention to the dynamic filter of each pixel.

Inspired by the dynamic filters described above, we propose a DySPN with a non-linear propagation model, which applies spatial and sequential attention to generate a series of adaptive affinity matrices.

**Our Approach**

In this paper, we propose DySPN for image-guided depth completion, including a naive non-linear propagation model, a diffusion suppression operation, and three variants that greatly reduce the complexities. The pipeline of our approach is demonstrated in Fig. 2.

**Naive Non-linear Propagation Model**

We design a non-linear propagation model (NLPM) that satisfies the desired requirements of SPN (Liu et al. 2017). Assuming \( h^0 \in \mathbb{R}^{m \times n} \) is the initial depth map and \( h^N \in \mathbb{R}^{m \times n} \) is the refined depth map after \( N \) iterations. \( \{h^b, b^{N}\} \) is reshaped as a column-first one-dimensional vector \( V^t \in \mathbb{R}^{mn} \), where \( t \in \{0, 1, 2 \cdots N\} \). The NLPM could be simply described as follows,

\[
V^{t+1} = GV^t
\]

\[
V^t = \begin{bmatrix}
1 - \lambda_{0,0} & w_{0,0}(1,0) & \cdots & w_{0,0}(m,n) \\
\vdots & \ddots & \ddots & \vdots \\
w_{m,0}(0,0) & \cdots & \cdots & 1 - \lambda_{m,n}
\end{bmatrix} V^t
\]

where \( G \in \mathbb{R}^{mn \times mn} \) is a fixed transformation matrix during the propagation, \( w_{i,j}(a,b) \) describes the affinity weight of pixel \((i,j)\) with its neighbors \((a,b)\) and \( \lambda_{i,j} = \sum_{a \neq i, b \neq j} w_{i,j}(a,b) \) means that the sum of each row of \( G \) is 1.

Ideally, our method adjusts the value of \( G \) by applying global attention. Therefore, the NLPM corresponds to the diffusion process with a partial differential equation (PDE) in Eq. (2).

\[
\begin{aligned}
V^{t+1} &= GV^t = (I - D^t + K^t \cdot A)V^t \\
V^{t+1} - V^t &= -(D^t - K^t \cdot A)V^t \\
\partial_t V^{t+1} &= -L^t V^t
\end{aligned}
\]

Where \( K^t \in \mathbb{R}^{mn \times mn} \) is the global attention matrix, \( D \) is the diagonal matrix that contains all the \( \lambda_{i,j} \), and \( A \) is the affinity matrix containing all the \( w_{i,j}(a,b) \).

\( L^t \) is the Laplace matrix that varies with the number of iterations \( t \), indicating that our model is non-linear. The transformation matrix \( G^t \) and the global attention matrix \( K^t \) could be concretely written as follows,

\[
G^t = \begin{bmatrix}
1 & \bar{w}_{0,0}^t(1,0) & \cdots & \bar{w}_{0,0}^t(m,n) \\
\bar{w}_{1,0}^t(0,0) & 1 - \bar{\lambda}_{1,0} & \cdots & \bar{w}_{1,0}^t(m,n) \\
\vdots & \ddots & \ddots & \vdots \\
\bar{w}_{m,n}^t(0,0) & \cdots & \cdots & 1 - \bar{\lambda}_{m,n}
\end{bmatrix}
\]

\[
K^t = \begin{bmatrix}
0 & \pi_{0,0}(1,0) & \cdots & \pi_{0,0}(m,n) \\
\pi_{1,0}^t(0,0) & 0 & \cdots & \pi_{1,0}^t(m,n) \\
\vdots & \ddots & \ddots & \vdots \\
\pi_{m,0}^t(0,0) & \cdots & \cdots & 0
\end{bmatrix}
\]

where \( \bar{w}_{i,j}^t(a,b) = \pi_{i,j}^t(a,b) w_{i,j}(a,b) \) is the weight of adaptive affinity, and \( \bar{\lambda}_{i,j} = \sum_{a \neq i, b \neq j} \bar{w}_{i,j}^t(a,b) \).

**Diffusion Suppression Operation**

The diffusion process would not stop until the end of \( N \) iterations, but this is inappropriate for all pixels. If \( N \) is too high, the output will be over-smoothing, which is negative for depth completion. To solve this problem, we adopt a diffusion suppression operation that estimates the affinity of refined depth maps in every iteration so that the model converges at an early stage.

We use a diagonal matrix \( \tilde{D}^t \) to describe the affinity of
refined depth maps between \( t \) and \( t + 1 \) step in Eq.(5).

\[
\bar{D}^t = I - D^t = \begin{bmatrix}
1 - \hat{\lambda}_{0,0} & 0 & \cdots & 0 \\
0 & 1 - \hat{\lambda}_{1,1} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 1 - \hat{\lambda}_{m,n}
\end{bmatrix}
\]

When the diffusion is completely convergent, the attention matrix \( K^t \) is a zero matrix, deducing \( \hat{\lambda}_{i,j} = 0 \). The diffusion process suppress as follows.

\[
V^{t+1} = G^t V^t = \bar{D}^t V^t = IV^t = V^t
\]

**Neighborhood decoupling of DySPN**

Our basic idea of \( N \)-step NLPM uses global attention with a large complexity of \( O(m^2n^2N) \). When we individually assign attention to each sampled neighbor, the complexity drops to \( O(k^2N) \). Meanwhile, the base network needs to estimate \( N \) times more feature maps than the linear propagation model, accompanied by a large number of additional calculations.

By analyzing the 7x7 CSPN affinity matrix, we find that the affinity weights of the neighbors at the same distance are correlated. Therefore, we came up with three variants for our DySPN, which decouple the neighborhood into parts regarding different distances. The complexity of our decoupled attention mechanism is only \( O(kN) \), which is irrelevant to image size \((m, n)\). Our method is easy to be implemented and lightly embedded in the propagation. As illustrated in Fig. 3, we reduce the amount of neighbors needed while achieving better accuracy in our experiment. Generally, one step propagation of our DySPN could be written as,

\[
h_{i,j}^{t+1} = \sum_{k \in Z_+} \sum_{(a,b) \in N_{i,j,k}^t} \pi_{i,j,k}^t \frac{w_{i,j}(a,b)}{S_{i,j}^t} h_{a,b}^t + \frac{\pi_{i,j,0}^t}{S_{i,j}^t} h_{i,j}^t + \left(1 - S_{i,j}^t\right) h_{i,j}^0
\]

where,

\[
S_{i,j} = \pi_{i,j,0} + \sum_{k \in Z_+} \sum_{(a,b) \in N_{i,j,k}^t} \pi_{i,j,k}^t w_{i,j}(a,b),
\]

\[
S_{i,j}^t = \pi_{i,j,0} + \sum_{k \in Z_+} \sum_{(a,b) \in N_{i,j,k}^t} \pi_{i,j,k}^t w_{i,j}(a,b).
\]

The affinity matrix \( A \) is usually extremely sparse because only a limited number of neighbors are sampled. Therefore, we use \( N_{i,j,k}^t \) to clearly represent its decoupled neighborhood. The basic form of \( N_{i,j,k}^t \) is defined as:

\[
N_{i,j,k}^t = \{(i + p, j + q)|p \in \{-2k + 1, 0, 2k - 1\}, q \in \{-2k + 1, 0, 2k - 1\}, (p, q) \neq (0, 0)\}
\]

At the bottom of Fig.3, Deformable DySPN has a trade-off between the number of neighbors and the size of receptive fields. Its neighborhood in pixel \((i, j)\) is defined as follows,

\[
N_{i,j,1}^t = \{(i + p, j + q)|p \in \{1, 0, -1\}, q \in \{1, 0, -1\}, (p, q) \neq (0, 0)\}
\]

\[
N_{i,j,k}^t = \{(i + p, j + q)|(p, q) \in f_\phi(k, (i,j)) \in \mathbb{R}, k \in \{2,3\}\}
\]

where \( f_\phi \) estimates the off-set of neighbors for pixel \((i,j)\).

**Network Architecture**

An Unet-like (Ronneberger, P.Fischer, and Brox 2015) encoder-decoder network with the backbone of ResNet-34 (He et al. 2016) is proposed as a base network for fair comparison with other SPNs. The head of the base network is modified to fit our DySPN, so that it outputs attention maps \( \pi_{i,j}^t \), an initial depth map \( h^0 \), an affinity matrix \( A \), and a confidence prediction of the sparse depth measurement in parallel. The confidence prediction is common in other works such as (Cheng et al. 2020a) (Cheng et al. 2020b) (Hu et al. 2021).
Table 1: Quantitative evaluation on NYU Depth v2 dataset. The CSPN, CSPN++, DeepLiDAR, FCFRNet, ACMNet, GuideNet, TWISE, NLSPN mean (Ma and Karaman 2018) (Cheng, Wang, and Yang 2018) (Park et al. 2020) (Liu et al. 2019) (Liu et al. 2021), (Zhao et al. 2021) (Tang et al. 2021) (Imran, Liu, and Morris 2021) (Park et al. 2020), respectively.

| Method       | RMSE (m) | REL | $\delta_{1.25}$ | $\delta_{1.25^2}$ | $\delta_{1.25^3}$ |
|--------------|----------|-----|-----------------|--------------------|--------------------|
| CSPN         | 0.117    | 0.016| 99.2            | 99.9               | 100.0              |
| CSPN++       | 0.116    | -    | -               | -                  | -                  |
| DeepLiDAR    | 0.115    | 0.022| 99.3            | 99.9               | 100.0              |
| FCFRNet      | 0.106    | 0.015| 99.5            | 99.9               | 100.0              |
| ACMNet       | 0.105    | 0.015| 99.4            | 99.9               | 100.0              |
| GuideNet     | 0.101    | 0.015| 99.5            | 99.9               | 100.0              |
| TWISE        | 0.097    | 0.013| 99.6            | 99.9               | 100.0              |
| NLSPN        | 0.092    | 0.012| 99.6            | 99.9               | 100.0              |
| 7×7 DySPN    | 0.090    | 0.012| 99.6            | 99.9               | 100.0              |
| Dilated -    | 0.091    | 0.012| 99.6            | 99.9               | 100.0              |
| Deformable - | 0.090    | 0.012| 99.6            | 99.9               | 100.0              |

$h^\text{gt}$ at pixel $(i, j)$; $\alpha = \beta = 1$ and $\rho \in \{1, 2\}$.

Experiment

In this section, we follow the same experiment settings with other SPN-based methods (i.e., perform experiments on KITTI DC and NYU Depth v2 dataset) to have a fair comparison. Ablation studies are presented to verify the effectiveness of our method.

Implementation Details

We implement our method on the PyTorch framework and train it with 4 NVIDIA RTX 2080 Ti GPUs. All of our experiments are trained by an ADAM optimizer with $\beta_1=0.9$, $\beta_2=0.999$. The learning rate starts at 0.001 for the first 30 epochs and reduces to 0.002 with another 10 epochs of training. Furthermore, we use a stochastic depth (Huang et al. 2016) strategy instead of the weight decay to prevent overfitting during training for better performance. For the NYU Depth V2 dataset, the batch size is set to 24 and 500 depth pixels were randomly sampled from the ground truth depth. For the KITTI DC dataset, the batch size is 8. Besides, data augmentation techniques are used, including horizontal random flip and color jitter.

Datasets

NYU Depth V2 dataset (Silberman et al. 2012) is comprised of RGB and depth images captured by a Microsoft Kinect camera in 464 indoor scenes. Following (Cheng, Wang, and Yang 2018) (Park et al. 2020) (Liu et al. 2021), our model is trained on a subset of 50K images from the official training split and tested on the 654 images from the official labeled test set. Each image is downsized to 320x240, and then 304x228 center-cropping was applied.

KITTI DC dataset (Uhrig et al. 2017) consists of over 90K RGB and LiDAR pairs for training, 1K pairs for validation, and another 1K pairs for testing. The training pairs are
Figure 5: Qualitative comparisons results with other methods on KITTI DC evaluation. (b) PENet (Hu et al. 2021), (c) ACMNet (Zhao et al. 2021), (d) FCFRNet (Liu et al. 2021).

### Table 2: Quantitative evaluation on KITTI DC Dataset

| Method       | RMSE (mm) | MAE (mm) | iRMSE (1/km) | iMAE (1/km) |
|--------------|-----------|----------|--------------|-------------|
| CSPN         | 1019.64   | 279.46   | 2.93         | 1.15        |
| TWISE        | 840.20    | 195.58   | 2.08         | 0.82        |
| DSPN         | 766.74    | 220.36   | 2.47         | 1.03        |
| DeepLiDAR    | 758.38    | 226.50   | 2.08         | 0.90        |
| ACMNet       | 744.91    | 206.09   | 2.08         | 0.90        |
| CSPN++       | 743.69    | 209.28   | 2.07         | 0.90        |
| NLSPN        | 736.24    | 218.83   | 2.25         | 0.99        |
| GuideNet     | 735.81    | 217.15   | 2.20         | 0.98        |
| FCFRNet      | 730.08    | 210.55   | 2.17         | 0.94        |
| PENet        | 709.12    | 192.71   | 1.88         | 0.82        |

For the NYU Depth V2 dataset, root mean square error (RMSE), mean absolute error (MAE), inverse RMSE (iRMSE), and inverse MAE (iMAE). All the metrics for evaluation are shown as follows.

\[
\begin{align*}
\text{RMSE}(mm) &= \sqrt{\frac{1}{v} \sum_x (\hat{h}_x - h_x)^2} \\
\text{MAE}(mm) &= \frac{1}{v} \sum_x |\hat{h}_x - h_x| \\
\text{iRMSE}(1/km) &= \sqrt{\frac{1}{v} \sum_x \left(\frac{1}{\hat{h}_x} - \frac{1}{h_x}\right)^2} \\
\text{iMAE}(1/km) &= \frac{1}{v} \sum_x \left|\frac{1}{\hat{h}_x} - \frac{1}{h_x}\right| \\
\text{REL} &= \frac{1}{v} \sum_x \left|\frac{h_x - \hat{h}_x}{h_x}\right| \\
\delta_\tau &= \max\left(\frac{h_x - \hat{h}_x}{h_x}\right) < \tau, \tau \in \{1.25, 1.25^2, 1.25^3\}
\end{align*}
\]

### Evaluation on NYU Depth v2 Dataset

We verified the effectiveness of our method in indoor scenes by evaluating on NYU Depth v2 Dataset. The quantitative comparison results are shown in Table 1. Our proposed algorithms achieve the best results and outperform other SoTA methods. Fig. 4 visualizes the attention maps learned by our network. Qualitatively, attention maps \( \pi^1_1 \) increase significantly with iteration, whereas attention maps \( \pi^2_1 \) and \( \pi^3_1 \) decrease. These attention maps, such as \( \pi^3_0 \) and \( \pi^3_2 \), contain a lot of boundary information. In contrast with CSPN++ (Cheng et al. 2020a), we find a majority of pixels use a big kernel with a large receptive field at first several iterations for fast recovery, while in the last iteration, pixels around object and surface boundary tend to use a smaller kernel, so that the attention map \( \pi^1_1 \) is becoming more and more obvious. We also observe an increasing trend in the attention maps of the diffusion suppression operation (i.e. \( \pi^2_1 \)).

### Evaluation on KITTI DC Dataset

We also verified the effectiveness of our method in outdoor scenes by evaluating on KITTI DC Dataset. The results of quantitative comparisons are shown in Table 2. We finally
submit the evaluation result of Deformable DySPN, which is the best of our three variations. Our DySPN outperforms all the other works with a significant improvement (nearly 21 mm in RMSE) and ranks 1st on KITTI DC evaluation in all metrics (RMSE, MAE, iRMSE, and iMAE) by the time of submission. In Fig. 5, we compare the results of our method with several works. Our work is more consistent in some cases where the depth maps are extremely sparse.

### Ablation Studies

In this section, we first perform ablation experiments to verify the effectiveness of our model design. Our experiments are trained on KITTI DC annotated train dataset and tested on the selected validation dataset. A simple 7×7 CSPN (Cheng, Wang, and Yang 2018) is implemented as our baseline model. Table 3 demonstrates the effectiveness of each design proposed in our 7×7 DySPN, including the DySPN module, the diffusion suppression (DS) operation, and the network training strategies. In addition, we employ a U-net with the backbone of ResNet-34 as our base network.

**DySPN module and CSPN module** are compared in 6 and 12 iterations respectively, as shown in Table 3. In particular, the RMSE is decreased from 811.0 mm of CSPN to 769.0 mm of DySPN with a significant 42 mm reduction when the number of iterations is 6. DySPN module also shows better convergence, because the RMSE is not affected when the number of iterations is increased from 6 to 12. We also find that there is little difference between the sigmoid layer and the softmax layer on our module.

**Diffusion suppression (DS) operation** is easy to be embedded to DySPN module as proposed in Eq. 7. As visualized in Fig. 4, high thermal values gradually appear at the boundary of the attention maps {π₀, · · · , π₆}, it clearly shows the edge-preserving effect of diffusion suppression. In Table 3, we observe a reduction in RMSE from 769.0 mm to 763.9 mm, which means our design is successful.

**Training strategy** of weight decay is mentioned in many depth completion works (Cheng et al. 2020a) (Hu et al. 2021) (Liu et al. 2021) (Tang et al. 2021) to reduce model over-fitting. In this case, we try different values of weight decay and figure out the best one as shown in Table 3. In addition, we adopt a stochastic depth strategy (Huang et al. 2016) to play a similar role as weight decay. Experiment results demonstrate that it is better to use stochastic depth alone than with weight decay.

**Dilated DySPN and Deformable DySPN** improve the effect and speed with fewer iterations and fewer neighbor samples, as illustrated in Table 4. Specifically, Dilated DySPN module is designed for a better speed-accuracy trade-off. It shows not only over 2× faster than the C2 module of PENet but also improves RMSE from 757.2 mm to 748.5 mm. Deformable DySPN is improved on precision. As summarized in Table 2, it ranks 1st on the KITTI DC benchmark at the time of submission.

### Conclusion

In this paper, we proposed DySPN with a non-linear propagation model, which applies spatial and sequential attention to generate a series of adaptive affinity matrices. Furthermore, we presented a diffusion suppression operation to avoid the over-smoothing problem. To reduce complexity, we elaborated on three variants, including 7×7 DySPN, Dilated DySPN, and Deformable DySPN, which have different speed-accuracy trade-offs. Our ablation studies verified the superior performance of our DySPN variations on both indoor and outdoor depth completion datasets. By the way, we believe that DySPN can not only be used for image-guided depth completion but also be applied to other aspects, such as image segmentation and cellular automata modeling.
