






## Article

# A Structurally Flexible Occupancy Network for 3-D Target Reconstruction Using 2-D SAR Images

Lingjuan Yu <sup>1</sup>, Jianlong Liu <sup>1</sup>, Miaomiao Liang <sup>1</sup>, Xiangchun Yu <sup>1</sup>, Xiaochun Xie <sup>2,\*</sup>, Hui Bi <sup>3</sup>  
and Wen Hong <sup>4</sup>

<sup>1</sup> Jiangxi Province Key Laboratory of Multidimensional Intelligent Perception and Control, School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China; yulingjuan@jxust.edu.cn (L.Y.)

<sup>2</sup> School of Physics and Electronic Information, Gannan Normal University, Ganzhou 341000, China

<sup>3</sup> College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

<sup>4</sup> Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100194, China

\* Correspondence: xiexiaochun@gnnu.edu.cn

**Abstract:** Driven by deep learning, three-dimensional (3-D) target reconstruction from two-dimensional (2-D) synthetic aperture radar (SAR) images has been developed. However, there is still room for improvement in the reconstruction quality. In this paper, we propose a structurally flexible occupancy network (SFONet) to achieve high-quality reconstruction of a 3-D target using one or more 2-D SAR images. The SFONet consists of a basic network and a pluggable module that allows it to switch between two input modes: one azimuthal image and multiple azimuthal images. Furthermore, the pluggable module is designed to include a complex-valued (CV) long short-term memory (LSTM) submodule and a CV attention submodule, where the former extracts structural features of the target from multiple azimuthal SAR images, and the latter fuses these features. When two input modes coexist, we also propose a two-stage training strategy. The basic network is trained in the first stage using one azimuthal SAR image as the input. In the second stage, the basic network trained in the first stage is fixed, and only the pluggable module is trained using multiple azimuthal SAR images as the input. Finally, we construct an experimental dataset containing 2-D SAR images and 3-D ground truth by utilizing the publicly available Gotcha echo dataset. Experimental results show that once the SFONet is trained, a 3-D target can be reconstructed using one or more azimuthal images, exhibiting higher quality than other deep learning-based 3-D reconstruction methods. Moreover, when the composition of a training sample is reasonable, the number of samples required for the SFONet training can be reduced.



check for updates

Academic Editor: Fabio Rocca

Received: 5 November 2024

Revised: 16 January 2025

Accepted: 17 January 2025

Published: 20 January 2025

**Citation:** Yu, L.; Liu, J.; Liang, M.; Yu, X.; Xie, X.; Bi, H.; Hong, W. A Structurally Flexible Occupancy Network for 3-D Target Reconstruction Using 2-D SAR Images. *Remote Sens.* **2025**, *17*, 347. <https://doi.org/10.3390/rs17020347>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** three-dimensional target reconstruction; 2-D SAR image; complex-valued attention mechanism; complex-valued long short-term memory; structurally flexible occupancy network

## 1. Introduction

Synthetic aperture radar (SAR) offers high-resolution imaging capabilities for all-day and all-weather conditions. Therefore, it has been widely used in civilian and military fields. For three-dimensional (3-D) SAR target reconstruction, radar can obtain both the 3-D geometric shapes and positions of targets, which are extremely useful for traffic control, urban management, military strikes, and rescue. Before deep learning technology was

used in SAR target reconstruction, tomography SAR (TomoSAR) [1], which enabled the high-resolution imaging of targets in the azimuth and height directions, was the focus. In recent years, 3-D target reconstruction based on deep learning has developed rapidly [2–4]. However, there are still some issues regarding 3-D target reconstruction based on TomoSAR and deep learning.

Tomographic SAR is mainly divided into multi-pass SAR and array SAR. Multi-pass SAR has made significant progress in regards to the platform, imaging algorithms, and imaging geometry. In terms of the platform, it has been expanded from airborne [5] and spaceborne [6,7] to unmanned aerial vehicles (UAV) [8,9]. In terms of imaging algorithms, they have been extended from employing spectral estimation [10] to the use of compressive sensing (CS) [11,12]. In terms of imaging geometry, the flight trajectory of radar has been extended from lines to circles [13–15]. Thanks to these advancements, multi-pass SAR has been applied for urban areas [16,17], forests [18–20], glaciers [21], etc. However, the increasing number of flights significantly increases the imaging time, leading to poor timeliness and decoherence. Array SAR can be further divided into down-looking array SAR and array interferometric SAR. In the early stages, some airborne down-view imaging systems were developed [22,23], and the corresponding imaging algorithms were also studied [24]. However, the down-looking array SAR suffered from a narrow mapping bandwidth and low cross-heading resolution. In the later stages, array interferometric SAR was extensively studied [25–32]. Based on multi-input and multi-output technology, it adopted cross-heading array antennas to generate multiple equivalent phase centers. Therefore, a single flight could obtain multi-channel data for 3-D imaging [26]. Array interferometric SAR has also been improved regarding the transmitted signal waveform, the channel numbers in the hardware system, imaging algorithms, imaging geometry, etc. The orthogonal frequency-division multiplexing chip waveform was used to avoid intra-pulse interferences [27]. The channels could be reduced to three when adopting an asymptotic 3-D phase unwrapping method [28]. The CS-based imaging algorithms obtained high-quality reconstruction results [29,30]. The flight trajectory of the radar could comprise two lines [31] or one circle [32]. Although array interferometric SAR offers good coherence and vital timeliness, the cost, weight, and complexity of the hardware system sharply increase as the number of channels increases.

Deep learning-based 3-D reconstruction technologies have been proposed to address the problems in TomoSAR. These could be divided into three categories according to their implementations and purposes in 3-D reconstruction. The first category involved post-processing the existing TomoSAR imaging results using the deep neural network to improve reconstruction quality [33]. This method undoubtedly increased imaging time. The second category involved replacing the 3-D TomoSAR imaging process with deep neural networks [34–37], model adaptive deep networks [38], etc. For example, deep neural networks were considered alternatives to the depth unfolding of the CS-based algorithms [34–37]. The disadvantage of these methods was that each range-azimuthal resolution unit was processed separately, without fully utilizing the structural characteristics of the target, unless some structural constraints were added to the network. In addition, each imaging process still required a large amount of multi-flight or multi-channel echo data. The third category involved directly reconstructing 3-D targets from two-dimensional (2-D) images, such as 3-D point generation networks [39,40], neural radiance fields [2–4], and pixel2mesh [41]. These networks extracted complete target features from 2-D SAR images. Once the deep model was trained, a 3-D target could be reconstructed from one or more images, without requiring multi-flight or multi-channel echo data. Obviously, the third category offers more advantages regarding the saving of hardware resources than do the other two categories.

In computer vision, numerous 3-D reconstruction methods were based on 2-D images. They were divided into explicit and implicit representation methods. The explicit representations further included voxel [42], point cloud [43], and mesh [44]. Voxel-based methods consumed a large amount of memory; point cloud-based methods could not represent the target surface; mesh-based methods could not be used for any topology structure. The implicit methods [45,46] involved learning a continuous mathematical function to determine whether a point in the target space belonged to the target. These methods could describe complex topological structures and continuous surfaces, thus obtaining high-quality reconstruction results. In addition, they required fewer parameters in the model training than did the explicit methods. The occupancy network (ONet) was an implicit method representing 3-D surfaces as continuous decision boundaries for the classifier [45]. It significantly reduced memory during the network training and allowed for arbitrary resolution reconstruction results by employing refinement methods in the subsequent inference process. In this paper, a structurally flexible occupancy network (SFONet) is proposed to reconstruct a 3-D target using one or more 2-D SAR images. We summarize our contributions as follows.

1. A SAR-tailored SFONet is proposed to reconstruct a 3-D target using one or more azimuthal images as the input. It includes a basic network and a pluggable module. In the basic network, a lightweight complex-valued (CV) encoder is designed to extract features from 2-D CV SAR images. The pluggable module is designed to include a CV long short-term memory (LSTM) submodule and a CV attention submodule. The former extracts structural features of the target from multiple azimuthal images, and the latter fuses these features.
2. A two-stage training strategy is also proposed when two input modes of the SFONet coexist. The basic SFONet is trained using one azimuthal image as the input, and then the pluggable module is trained using multiple azimuthal images as the input. This strategy saves training time and allows the second stage to focus on mining the target structure information implied in multiple azimuthal SAR images.
3. One dataset containing 2-D images and 3-D ground truth is constructed using the Gotcha echo dataset. Comparative experiments with other deep learning methods and ablation experiments are implemented. The number of CV LSTM layers and refinement times in the reference are also analyzed. Additionally, the roles of CV LSTM and CV attention and the composition of the training samples are discussed.

The remainder of this paper is organized as follows. Section 2 introduces related works concerning the original ONet and LSTM. Section 3 presents the detailed structure of the SFONet and illustrates the processes of model training and inference. Section 4 implements the experiment and analyzes the experimental results. Section 5 discusses the roles of the pluggable module and the composition of the training samples. Finally, Section 6 provides the conclusion.

## 2. Related Work

### 2.1. Occupancy Network

The ONet was first proposed by Mescheder et al. [45]. It represents the surface reconstruction of a 3-D target as the continuous decision boundaries for a binary classifier. It does not predict an explicit representation with a fixed resolution. Instead, it predicts the probability of being occupied by the target for each point  $p \in R^3$  sampled from a unified target space, significantly reducing memory. In the training stage, the ONet can be approximated as an occupancy function  $f_\theta$ , where  $\theta$  represents the network parameters. The function's inputs usually include the observed values  $x \in \mathcal{X}$  and the point position  $p \in R^3$ . They can also be abbreviated as  $(p, x) \in R^3 \times \mathcal{X}$ . The function's output is an RV

number between 0 and 1, representing the occupancy probability. Therefore, the occupancy function can be represented by  $f_{\theta} : R^3 \times \mathcal{X} \rightarrow [0, 1]$ . In the inference, one unit space is discretized into voxel units at a specific resolution. Then, the ONet is used to evaluate the occupancy of vertices for each voxel unit. Furthermore, the octree algorithm [47] can be used in refining the voxels, and the ONet evaluates the newly added vertices of the refined voxels until the required resolution is achieved. At last, the marching cubes algorithm [48] can extract a smooth 3-D surface mesh.

Recently, various variants of ONet have emerged. The convolutional ONet allowed for the 3-D reconstruction of a large-scale scene rather than a single target [49]. The dynamic plane convolutional ONet enabled the learned dynamic plane to capture rich features in the direction of maximum information for 3-D surface reconstruction [50]. So far, the ONet has been applied to the 3-D building reconstruction from a 2-D satellite image [51]. To our knowledge, the ONet has not been used in 3-D SAR target reconstruction.

## 2.2. LSTM

LSTM is a type of time-recursive network (RNN) [52]. It better handles long-term dependencies in sequences by introducing one cell state and three gates (i.e., forget gate, input gate, and output gate). The forget gate controls the amount of forgotten information from the previous cell; the input gate controls the amount of information newly added to the current cell; the output gate controls the amount of output information. LSTM has been successfully used in language, speech, and images, capturing the long-time dependencies of these signals.

In recent years, LSTM has been extensively applied in SAR image interpretation. For change detection based on interferometric SAR time series, LSTM captured changes in urban areas and volcanoes [53]; bidirectional LSTM (Bi-LSTM) achieved anomaly detection and the classification of sinkholes [54]. For moving target detection (MTD), Bi-LSTM suppressed the missing alarms by tracking the shadows of targets [55], and the trajectory smoothing LSTM helped to refocus the ground-moving targets [56]. For target recognition, LSTM helped to improve the classification performance by fusing features extracted from SAR images with adjacent azimuthal angles [57] or learning the long-term dependent features from sequence SAR images [58]. For the land cover classification based on polarimetric SAR (PolSAR) images, LSTM was used to obtain spatial-polarimetric features in adjacent pixels to improve classification accuracy [59].

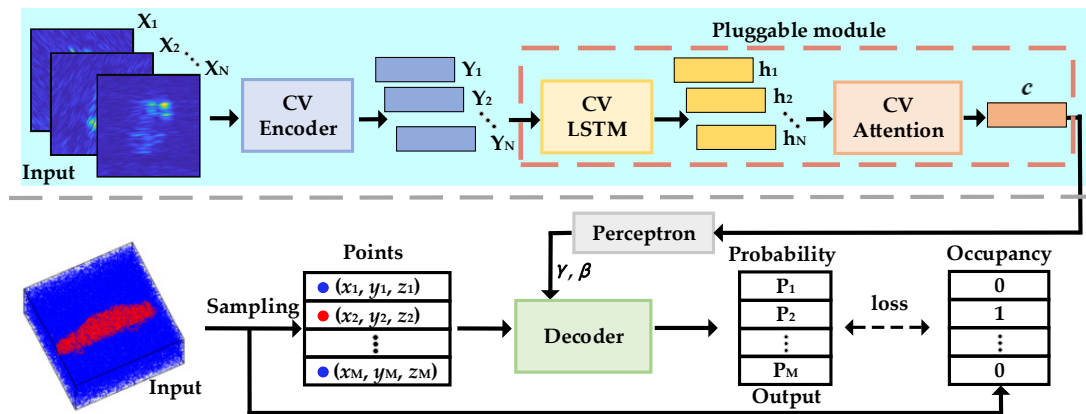
## 3. Methodology

We first present the overall architecture of the proposed SFONet. Then, we introduce the detailed structure of each module in this framework, illustrate the detailed training process, and provide the inference process from 2-D SAR images to a 3-D target.

### 3.1. Framework of the Structurally Flexible Occupancy Network

The framework of the SFONet is shown in Figure 1. We divide it into two branches using the gray dashed line in the middle. The upper branch is used for feature extraction from 2-D SAR images, while the lower branch is used to predict a probability set of points sampled from the unit target space. The upper branch includes a lightweight CV encoder and a pluggable module (in the orange dashed box). Furthermore, the pluggable module comprises a CV LSTM submodule and a CV attention submodule. When the input is one azimuthal image, only the CV encoder extracts features. The pluggable module is used for further feature extraction and fusion when the inputs are multiple azimuthal images. We uniformly denote the inputs of this branch as  $\{X_1, X_2, \dots, X_n, \dots, X_N\}$  ( $N \geq 1$ ) and the output feature vector as  $c$ . The lower branch includes an RV perceptron and an RV decoder. The input of the RV perceptron is the encoded feature vector  $c$ , and the outputs

are parameters  $\gamma$  and  $\beta$  used for the CBN of the RV decoder. For the RV decoder, the inputs are random sampling points in the unit target space, some of which come from the target (represented by red dots), and the others are outside the target (represented by blue dots). The whole sampling point set is denoted as  $\{p_1, p_2, \dots, p_m, \dots, p_M\}$  ( $M \gg 1$ ), and the 3-D coordinates of each sampling point  $p_m$  as  $(x_m, y_m, z_m)$ . The output of the RV decoder is the occupancy probability set  $\{P_1, P_2, \dots, P_m, \dots, P_M\}$ , where each element has a value between 0 and 1. Additionally, the true occupancy probability of each sampling point is either 0 (for a blue dot) or 1 (for a red dot).

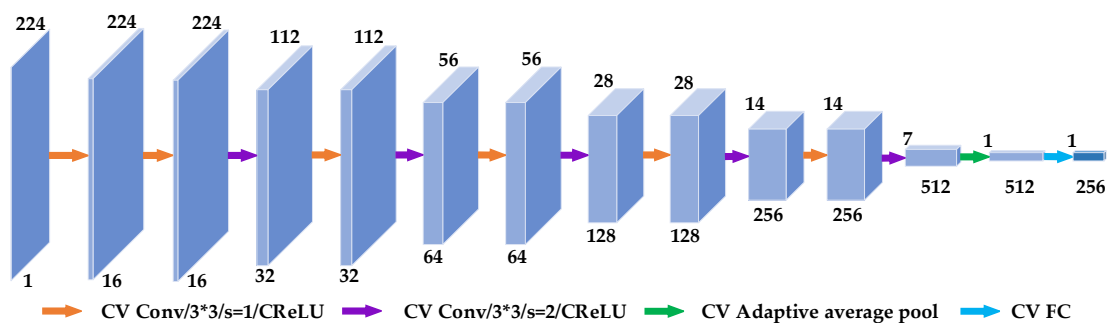


**Figure 1.** Framework of SFONet. It includes a CV encoder, a pluggable module (consisting of a CV LSTM submodule and a CV attention submodule), an RV perceptron, and an RV decoder.

From another perspective, we divide the SFONet into a basic network and a pluggable module. When the input is one azimuthal image, the basic network is used. It includes a CV encoder, an RV perceptron, and an RV decoder. When the input is the composition of multiple azimuthal images, the pluggable module is added to the basic network. Compared with the original ONet, improvements in the SFONet (in the background light-blue area) are as follows. (1) The input of SFONet can be one or more 2-D images. (2) The encoder is lightweight and extends to the CV domain, suitable for small CV SAR datasets. (3) A pluggable module is designed to extract the structural features of the target from multiple azimuthal SAR images.

### 3.2. CV Encoder

Unlike optical image datasets, SAR datasets are always small. Moreover, CV SAR data contain amplitude and phase information. Therefore, we design a lightweight CV encoder to extract features from one or multiple azimuthal SAR images. As shown in Figure 2, the operations include convolution, adaptive average pooling, full connection (FC), and ReLU. The FC operation further includes multiplication and addition. Although all these operations are CV, they can be converted to the corresponding RV operations.



**Figure 2.** Architecture of CV encoder.

$F$  is a CV feature vector,  $W$  is a CV weight vector, and  $b$  is a CV bias. Then, the CV operations mentioned above can be represented by

$$\begin{cases} W * F + b = \Re(W) * \Re(F) - \Im(W) * \Im(F) + \Re(b) + i(\Re(W) * \Im(F) + \Im(W) * \Re(F) + \Im(b)) \\ W \cdot F + b = \Re(W) \cdot \Re(F) - \Im(W) \cdot \Im(F) + \Re(b) + i(\Re(W) \cdot \Im(F) + \Im(W) \cdot \Re(F) + \Im(b)) \\ CReLU(F) = ReLU(\Re(F)) + i(ReLU(\Im(F))) \\ CAdaptAvgPool(F) = AdaptAvgPool(\Re(F)) + i(AdaptAvgPool(\Im(F))) \end{cases} \quad (1)$$

where  $*$  denotes the convolution operation,  $CReLU(\cdot)$  denotes the CV ReLU activation, and  $CAdaptAvgPool(\cdot)$  and  $AdaptAvgPool(\cdot)$  denote the CV and RV adaptive average pooling operations, respectively.

### 3.3. CV LSTM

Multiple azimuthal SAR images contain richer structural information about the target than does a single SAR image. Thus, we design a two-layer CV LSTM to extract the structural features of the target from multiple azimuthal SAR images. As shown in Figure 3a, the input of CV LSTM is one azimuthal feature set  $\{Y_1, Y_2, \dots, Y_n, \dots, Y_N\}$ , obtained through the CV encoder, and the final output is a sequence feature set  $\{h_1, h_2, \dots, h_n, \dots, h_N\}$ . The architecture of each CV LSTM cell is shown in Figure 3b, which is the same as that of the RV LSTM. There are also three gates: the forget gate, the input gate, and the output gate. Unlike the RV LSTM, all operations involved in these gates are CV.

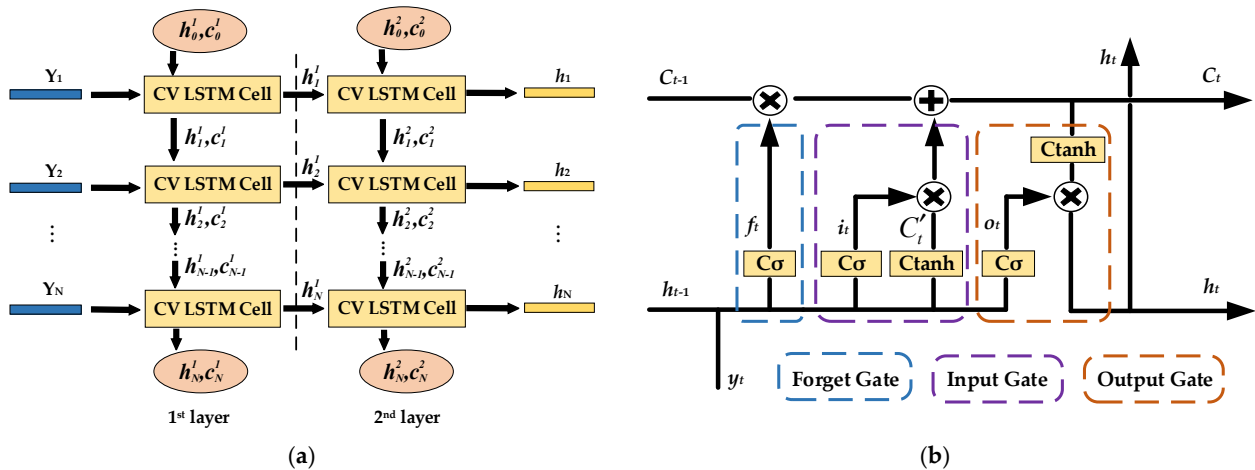


Figure 3. (a) A two-layer CV LSTM. (b) Architecture of a CV LSTM cell.

To avoid redundant introductions of some CV operations, we list the following common CV operations involved in the three gates: multiplication, addition, and activation. Since CV multiplication and addition are given in (1), here, we only define two CV activation functions, as follows:

$$\begin{cases} C\sigma(\cdot) = \sigma(\Re(\cdot)) + i\sigma(\Im(\cdot)) \\ Ctanh(\cdot) = \tanh(\Re(\cdot)) + itanh(\Im(\cdot)) \end{cases} \quad (2)$$

where  $C\sigma(\cdot)$  is a CV sigmoid function;  $Ctanh(\cdot)$  is a CV tanh function;  $\sigma(\cdot)$  and  $\tanh(\cdot)$  are their corresponding RV functions.

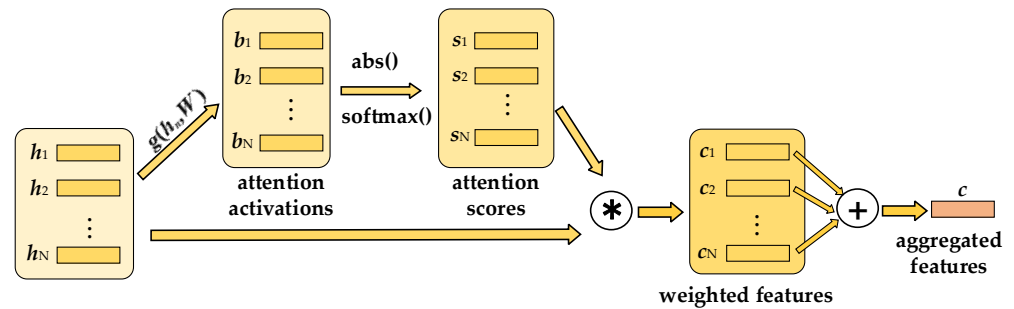
Then, all the CV operations involved in the three gates can be represented by

$$\begin{cases} f_t = C\sigma(W_f \cdot [h_{t-1}, y_t] + b_f) \\ i_t = C\sigma(W_i \cdot [h_{t-1}, y_t] + b_i) \\ C'_t = C\tanh(W_c \cdot [h_{t-1}, y_t] + b_c) \\ C_t = f_t \cdot C_{t-1} + i_t \cdot C'_t \\ o_t = C\sigma(W_o \cdot [h_{t-1}, y_t] + b_o) \\ h_t = o_t \cdot C\tanh(C_t) \end{cases} \quad (3)$$

where  $f_t$ ,  $i_t$  and  $o_t$  are the CV control signals of the forget gate, the input gate, and the output gate, respectively;  $C_{t-1}$  and  $C_t$  are the CV states of the previous cell and the current cell, respectively;  $C'_t$  is the new CV candidate value for the current cell state;  $W_f$ ,  $W_i$ ,  $W_c$ , and  $W_o$  are the learnable CV weights;  $b_f$ ,  $b_i$ ,  $b_c$ , and  $b_o$  are the learnable CV biases;  $h_t$  is the current output.

### 3.4. CV Attention

As shown in Figure 3a, the last time step output  $h_N$  of CV LSTM integrates more azimuthal information than the other time step outputs. However, other time step outputs still contain additional information that the last time step output does not possess. Inspired by Ref. [60], the CV attention submodule shown in Figure 4 is designed to fuse features from all the time steps. In this submodule, we sequentially obtain attention activations, attention scores, weighted features, and aggregated features. The detailed process is as follows:



**Figure 4.** Architecture of CV attention.

First, we obtain CV attention activation values. The input set  $\mathcal{A} = \{h_1, h_2, \dots, h_n, \dots, h_N\}$  is from the CV encoder. For each feature vector  $h_n \in \mathbb{C}^{1 \times D}$  ( $D$  is the feature dimension), the attention activation  $b_n$  can be calculated through a CV FC with the shared weight  $W$ ,

$$b_n = g(h_n, W) = h_n W \quad (4)$$

where  $W \in \mathbb{R}^{D \times D}$ ,  $b_n \in \mathbb{R}^{1 \times D}$ , and  $g$  represents CV FC.

Second, we obtain CV attention scores. Considering that  $b_n = [b_n^1, b_n^2, \dots, b_n^d, \dots, b_n^D]$  is a CV vector, we take the modulus and then use the softmax function to normalize the  $d$ -th ( $d = 1, 2, \dots, D$ ) dimensional feature value. The  $d$ -th dimensional value  $s_n^d$  of the attention score vector  $s_n = [s_n^1, s_n^2, \dots, s_n^d, \dots, s_n^D]$  can be expressed by

$$s_n^d = \frac{e^{|b_n^d|}}{\sum_{j=1}^N e^{|b_j^d|}} \quad (5)$$

where  $|\cdot|$  represents the modulus operation.

Third, the original feature vector  $h_n$  is weighted by the corresponding attention score vector  $s_n$  on each dimension to yield a new feature vector  $c_n$ . It can be expressed by

$$c_n = h_n \cdot s_n \tag{6}$$

where  $c_n = [c_n^1, c_n^2, \dots, c_n^d, \dots, c_n^D]$ .

Finally, the weighted features are summed up on each feature dimension to obtain the aggregated feature vector, and the modulus is calculated to match the subsequent RV perceptron. The final output is denoted as  $c = [c^1, c^2, \dots, c^d, \dots, c^D]$  ( $c \in \mathbb{C}^{1 \times D}$ ), and then the  $d$ -th dimensional value of the output  $c$  can be written by

$$c^d = \left| \sum_{n=1}^N c_n^d \right| \tag{7}$$

### 3.5. RV Perceptron and Decoder

The architectures of both the RV perceptron and the RV decoder are shown in Figure 5. For the RV perceptron, the main operation is a one-dimensional (1-D) convolution. Its input is the encoded feature vector  $c$ , and its outputs are the parameters  $\gamma$  and  $\beta$ , which are used for CBN in the RV decoder. The RV decoder mainly consists of five residual blocks, and the operations in each block comprise 1-D convolution, CBN, ReLU, and addition. Beyond these residual blocks, one CBN, 1-D convolution, and sigmoid operation exist at the end. The inputs of the RV decoder are the 3-D coordinate positions of the sampling points in the unit target space, and the output is a set of occupancy probabilities for all the sampling points.

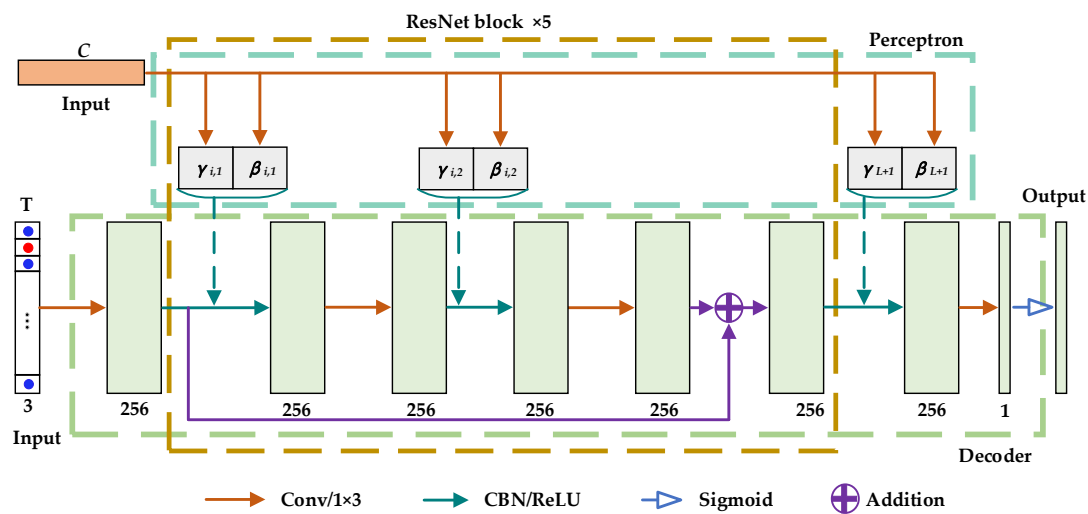


Figure 5. Architectures of both the RV perceptron and the RV decoder.

The CBN mentioned above guides good 3-D target reconstruction using the aggregated feature vector  $c$ . The calculation formula can be expressed by

$$F_o = \gamma \frac{F_i - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{8}$$

where  $F_i$  and  $F_o$  represent the features before and after CBN, respectively;  $\mu$  and  $\sigma^2$  represent the mean and variance of  $F_i$ , respectively;  $\epsilon$  represents an arbitrarily small number close to 0.

After the last 1-D convolution operation in Figure 5, we can obtain the final output feature vector  $F_{final}$ , with a size of  $1 \times M$ .  $F_{final}^m$  is the feature value of the  $m$ -th ( $m = 1,$



$2, \dots, M$ ) sampling point. Then, the sigmoid function is used to calculate the occupancy probability  $P_m$ . The calculation formula can be expressed by

$$P_m = \frac{1}{1 + \exp(-F_{final}^m)} \quad (9)$$

### 3.6. Training

A two-stage strategy is proposed for the training of the SFONet when its two input modes coexist. The basic network is trained first, and then the pluggable module is trained. This strategy not only saves time but also improves performance. The reasons for the improvement can be explained in the two stages. In the first stage, one azimuthal image corresponds to a 3-D ground truth, resulting in no blurring. The second stage can focus on extracting the structural features from multiple azimuthal images and performing feature fusion.

In the first stage, the parameter set of the basic network is  $\Theta_{base}$ , the batch size is  $B$ , the  $i$ -th ( $i = 1, 2, \dots, B$ ) image in a training batch is  $X_i$ , and  $f$  is the mapping function of the basic network. For the  $j$ -th ( $j = 1, 2, \dots, M$ ) sampling point in the  $i$ -th target space, it is denoted as  $p_{ij}$  and its true occupancy probability as  $O_{ij}$ . Then, the cross-entropy loss function can be calculated by

$$\mathcal{L}_1(\Theta_{base}) = \frac{1}{|B|} \sum_{i=1}^{|B|} \sum_{j=1}^M \mathcal{L}(f_{\Theta_{base}}(p_{ij}, X_i), O_{ij}) \quad (10)$$

The parameter set  $\Theta_{base}$  is updated by ascending the stochastic gradient  $\nabla_{\Theta_{base}} \mathcal{L}_1(\Theta_{base})$ .

In the second stage, the parameter set  $\Theta_{base}$  is fixed. The parameter set of the pluggable module is  $\Theta_{LSTM+Atten}$ , the  $i$ -th ( $i = 1, 2, \dots, B$ ) image set in a training batch is  $\mathcal{X}_i = \{X_{i1}, X_{i2}, \dots, X_{iN}\}$ , and  $f'$  is the mapping function of the SFONet with the pluggable module added. Then, the cross-entropy loss function is rewritten as

$$\mathcal{L}_2(\Theta_{LSTM+Atten}) = \frac{1}{|B|} \sum_{i=1}^{|B|} \sum_{j=1}^M \mathcal{L}(f'_{\Theta_{LSTM+Atten}}(p_{ij}, \mathcal{X}_i, \Theta_{base}), O_{ij}) \quad (11)$$

The parameter set  $\Theta_{LSTM+Atten}$  is updated by ascending the stochastic gradient  $\nabla_{\Theta_{LSTM+Atten}} \mathcal{L}_2(\Theta_{LSTM+Atten})$ .

The two-stage training process can be summarized in Algorithm 1. The calculation formulas of the training loss can also apply to the validation loss. Since these formulas are not divided by the number of points, the maximum value of the training or validation loss may be greater than 1. In Section 4.4, we provide the validation loss curves.

---

#### Algorithm 1 Two-Stage Training

---

##### Stage 1:

**Inputs:** Batch size  $B$ , Number of sampling points  $M$   
**for** the number of iterations **do**

- Choose images  $\{X_1, \dots, X_i, \dots, X_B\}$  as a batch, sample  $M$  points in the  $i$ -th ( $i = 1, 2, \dots, B$ ) 3-D unit target space, and obtain the true occupancy probability  $O_{ij}$  of the  $j$ -th ( $j = 1, 2, \dots, M$ ) sampling point.
- Predict the occupancy probability of the sampling point  $p_{ij}$ .
- Calculate the loss function in (10) and update the parameter set  $\Theta_{base}$ .

**Output:**  $\Theta_{base}$

---

**Algorithm 1** Cont.**Stage 2:**

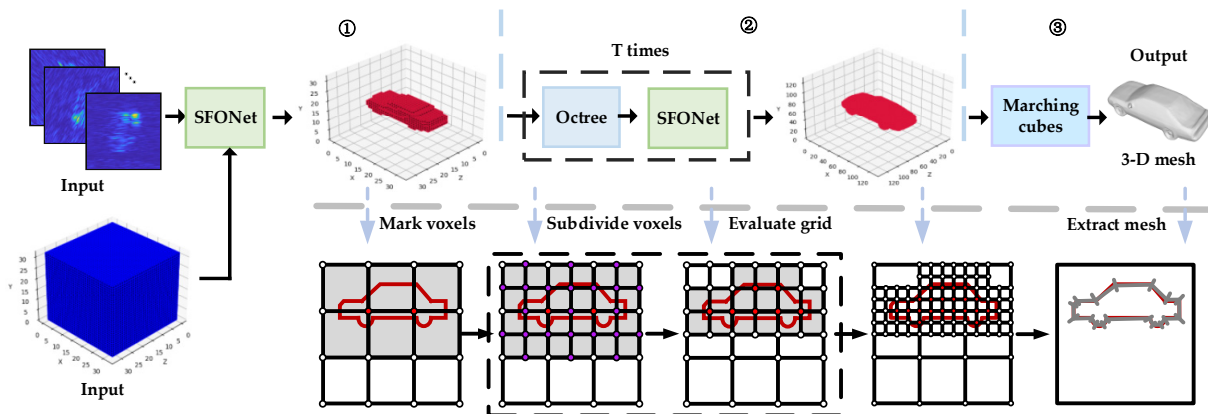
**Inputs:** Batch size  $B$ , Number of sampling points  $M$ , Parameter set  $\Theta_{base}$   
**for** the number of iterations **do**

- Choose image sets  $\{\mathcal{X}_1, \dots, \mathcal{X}_i, \dots, \mathcal{X}_B\}$  as a batch, sample  $M$  points in the  $i$ -th ( $i = 1, 2, \dots, B$ ) 3-D unit target space, and obtain the true occupancy probability  $O_{ij}$  of the  $j$ -th ( $j = 1, 2, \dots, M$ ) sampled point.
- Predict the occupancy probability of the sampling point  $p_{ij}$ .
- Calculate the loss function in (11) and update the parameter set  $\Theta_{LSTM+Atten}$ .

**Output:**  $\Theta_{LSTM+Atten}$

**3.7. Inference**

The inference uses the trained SFONet to reconstruct a 3-D target from a test sample. The composition of a test sample is the same as that of a training sample, which can be one or multiple azimuthal images. Figure 6 shows the detailed inference process. It includes three steps: the initial 3-D reconstruction of the target, the refinement of the reconstructed target, and the mesh extraction.



**Figure 6.** Inference process.

In the first step, a unit volume space, different from the unit target space in the training stage, is discretized into multiple voxels at an initial resolution, and the vertices of each voxel are also called grid points. Then, the SFONet  $f_{\Theta}(p, \mathbf{X})$  is used to evaluate the occupancy status of each grid point  $p$ . If  $f_{\Theta}(p, \mathbf{X}) \geq \tau$  ( $\tau$  is a threshold), mark the grid point as occupied; otherwise, mark it as unoccupied. When a voxel has more than two adjacent grid points with different occupied states, this voxel is marked as active, which means that this voxel may be the boundary location of the target. All the activated voxels form the initial 3-D reconstruction target.

In the second step, the initial reconstruction results are refined using the classic octree algorithm. Each activated voxel in the first step is subdivided into eight sub-voxels. Then,  $f_{\Theta}(p, \mathbf{X})$  is used to evaluate the occupancy probability of each new grid point for each sub-voxel. All the newly activated sub-voxels form the refined target. Repeat the occupancy evaluation of the new grid points and the subdivision of voxels  $T$  times until the desired resolution is achieved.

In the final step, the marching cubic algorithm extracts the final 3-D mesh, since the mesh representation exhibits a better visualization effect than does the voxel. This extraction obtains the approximate iso-surfaces by using  $\{p \in \mathbb{R}^3 | f_{\Theta}(p, \mathbf{X}) = \tau\}$ .

The inference process can be summarized in Algorithm 2.

---

**Algorithm 2** Inference

---

**for** the number of test samples **do**

- Discretize a unit cube into voxels, and the vertices of each voxel are grid points.
  - Evaluate the occupancy probability of each grid point  $p$  by using the SFONet  $f_{\Theta}(p, X)$ . If  $f_{\Theta}(p, X) \geq \tau$  ( $\tau$  is a threshold),  $p$  is marked as occupied; otherwise, it is marked as unoccupied.
  - Mark the activated voxel and form an initial 3-D reconstruction target.  
**for** the number of refinement iterations **do**
    - Use the octree algorithm to subdivide each activated voxel into eight sub-voxels, producing new grid points.
    - Evaluate the occupancy probability of each new grid point.
    - Mark the activated sub-voxel and form a refined 3-D reconstruction target.
  - Use the marching cubes algorithm to extract approximate iso-surfaces under the condition  $\{p \in \mathbb{R}^3 | f_{\Theta}(p, X) = \tau\}$ , and obtain the final 3-D target mesh.
- 

## 4. Experiments and Analysis

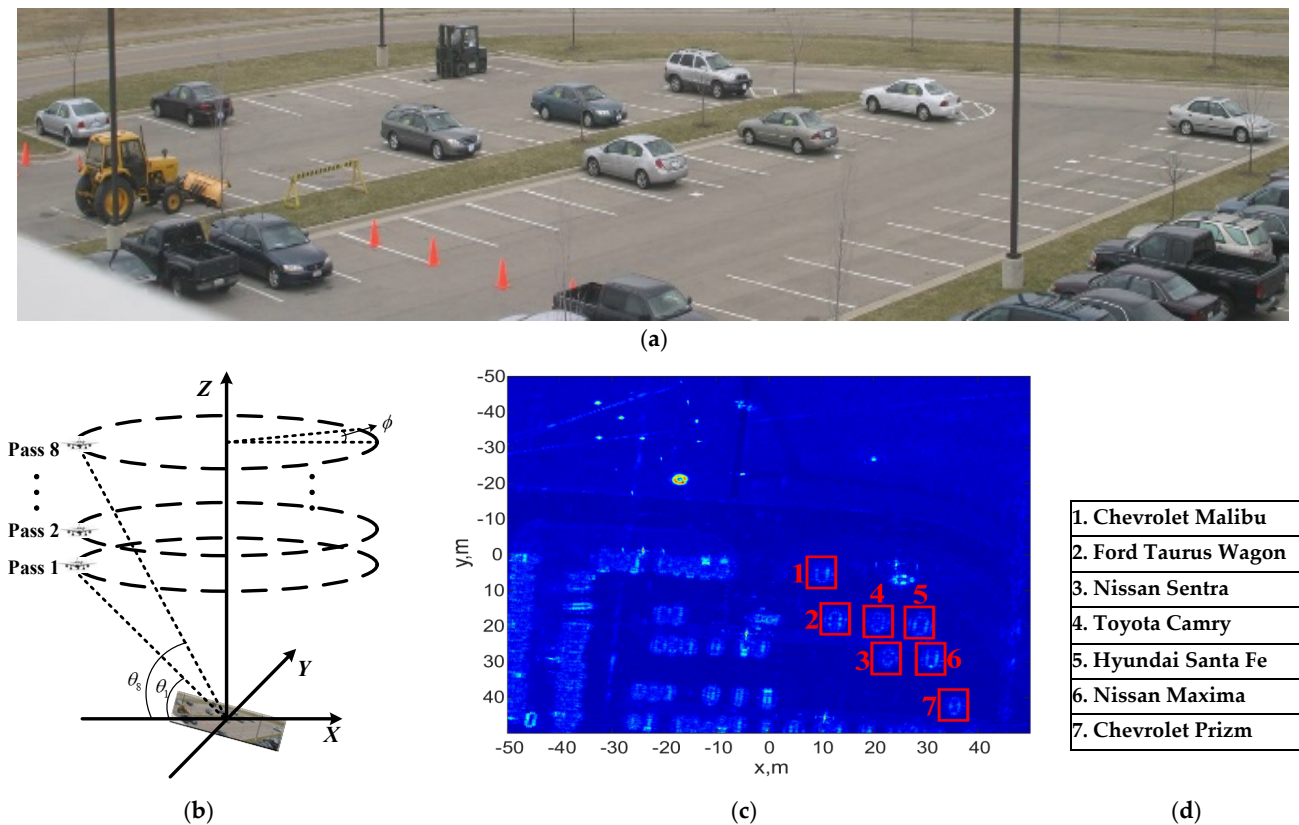
We construct a dataset that includes 2-D images and 3-D ground truth using the Gotcha echo data. Then, we conduct comparative experiments, ablation experiments, and experiments on parameters such as the number of CV LSTM layers and the number of refinement iterations in the inference.

### 4.1. Dataset

The Gotcha echo data [61] were collected by the airborne fully polarimetric radar system. The observed area was a parking lot, where many civilian cars and industrial vehicles were parked, as shown in Figure 7a. The imaging geometry is shown in Figure 7b, where the radar flew eight passes around the target area in a circular trajectory. During the data collection on each pass, the azimuthal interval is about 1 degree. We use the HH polarimetric SAR data collected from the first circular pass for imaging. The imaging result is shown in Figure 7c.

Aiming to construct a 2-D image dataset for 3-D reconstruction, we divide the whole aperture of each pass into 36 sub-apertures, each with an azimuthal angle of 10 degrees. With these sub-aperture echoes, 1152 sub-aperture images of the entire scene are obtained. Then, we crop slices of seven cars for each sub-aperture image, with the brands shown in Figure 7d. Finally, a total of 8064 slices can be obtained, each containing only one car. To form the 3-D ground truth, we download CAD models of seven vehicles from publicly available dataset websites and normalize their size.

In the following experiments, the data portioning is listed in Table 1. The training set includes 3024 slice images of seven cars from the first three passes. The validation set includes 1008 slice images from the fourth pass. The test set includes 4032 slice images from the fifth to the eighth pass. It is worth noting that the two input modes of the SFONet share the same data partitioning, but the compositions of their training samples are different. For the single-input mode, each training sample includes only one azimuthal slice image. For the multiple-input mode, each training sample comprises multiple slice images from the same pass but different azimuthal angles. In addition, the composition of each validation or test sample is kept consistent with the corresponding training sample in the two input modes.



**Figure 7.** (a) Optical photos of a parking lot; (b) imaging geometry; (c) 2-D SAR imaging result; (d) brands of seven cars.

**Table 1.** Data partitioning.

	Training Set	Validation Set	Test Set
Pass number	1, 2, 3	4	5, 6, 7, 8
Number of images	3024	1008	4032

#### 4.2. Implement Details and Evaluation Metrics

All the experiments are implemented on a computer with an Intel Xeon(R) W-2235 CPU and an NVIDIA GeForce RTX 3090 GPU. The computer's memory is 64 GB, and the graphics memory is 24 GB. The operating system is Ubuntu 20.04, the deep learning framework is Pytorch 1.12.1, and the programming language is Python 3.8.16. The libraries CUDA 11.3 and CUDANN 8.2.0 are used. During the network training, the Adam optimizer is used. The batch size is set to 16, and the learning rate is set to  $10^{-4}$ .

We use intersection and union ratio (IoU), chamfer-L1 distance (CD), and normal consistency (NC) [45] to evaluate the reconstruction performance quantitatively. Since both the predicted and ground truth mesh are normalized in the 3-D unit space, we randomly sample some points in the space. Suppose that  $\mathcal{M}_{\text{pred}}$  and  $\mathcal{M}_{\text{GT}}$  represent sets of sampling points inside or on the surfaces of the predicted and ground truth mesh, respectively. Then, the volume IoU can be calculated by

$$\text{IoU}(\mathcal{M}_{\text{pred}}, \mathcal{M}_{\text{GT}}) = \frac{|\mathcal{M}_{\text{pred}} \cap \mathcal{M}_{\text{GT}}|}{|\mathcal{M}_{\text{pred}} \cup \mathcal{M}_{\text{GT}}|} \quad (12)$$

Assume that  $\partial\mathcal{M}_{\text{pred}}$  and  $\partial\mathcal{M}_{\text{GT}}$  represent the surfaces of the predicted mesh and the ground truth mesh, respectively.  $p$  and  $q$  represent sampling points on  $\partial\mathcal{M}_{\text{pred}}$  and  $\partial\mathcal{M}_{\text{GT}}$ ,

respectively.  $n(p)$  and  $n(q)$  represent the normal vectors on  $\partial\mathcal{M}_{\text{pred}}$  and  $\partial\mathcal{M}_{\text{GT}} \cdot \text{proj}_1(q)$  and  $\text{proj}_2(p)$  represent projections of  $q$  and  $p$  on  $\partial\mathcal{M}_{\text{pred}}$  and  $\partial\mathcal{M}_{\text{GT}}$ , respectively.  $\langle \bullet, \bullet \rangle$  represents the inner product. Then, the CD and Nc can be calculated by

$$CD(\mathcal{M}_{\text{pred}}, \mathcal{M}_{\text{GT}}) = \frac{1}{2|\partial\mathcal{M}_{\text{pred}}|} \int_{\partial\mathcal{M}_{\text{pred}}} \min_{q \in \partial\mathcal{M}_{\text{GT}}} \|p - q\| dp + \frac{1}{2|\partial\mathcal{M}_{\text{GT}}|} \int_{\partial\mathcal{M}_{\text{GT}}} \min_{p \in \partial\mathcal{M}_{\text{pred}}} \|p - q\| dq \quad (13)$$

$$NC(\mathcal{M}_{\text{pred}}, \mathcal{M}_{\text{GT}}) = \frac{1}{2|\partial\mathcal{M}_{\text{pred}}|} \int_{\partial\mathcal{M}_{\text{pred}}} |\langle n(p), n(\text{proj}_2(p)) \rangle| dp + \frac{1}{2|\partial\mathcal{M}_{\text{GT}}|} \int_{\partial\mathcal{M}_{\text{GT}}} |\langle n(\text{proj}_1(q), n(q)) \rangle| dq \quad (14)$$

The mean IoU (mIoU), mean CD (mCD), and mean NC (mNC) represent the average IoU, CD, and NC of multiple reconstructed targets, respectively.

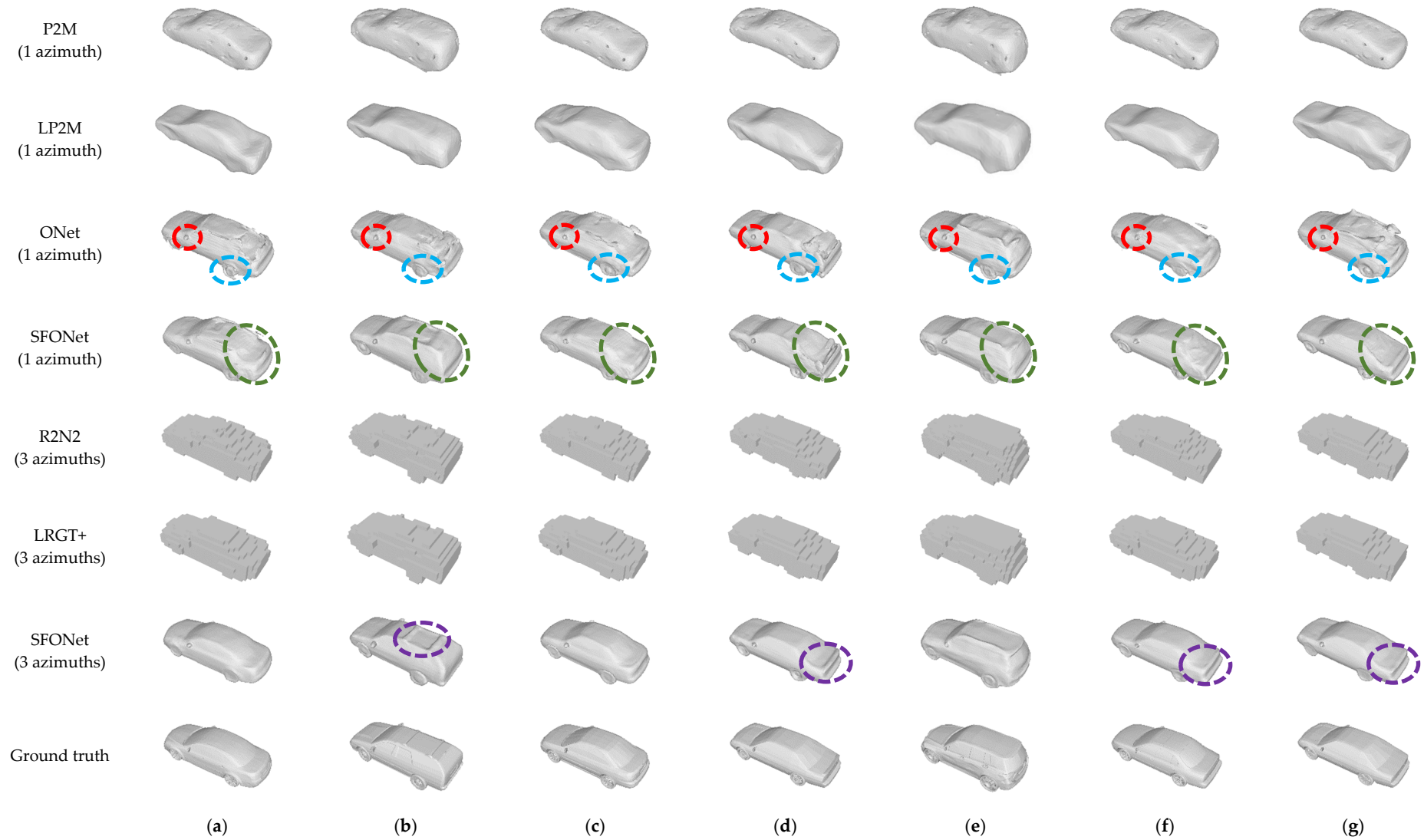
#### 4.3. Comparative Experiments

We compare two input modes of the SFONet. Each training or test sample has only one azimuthal image in one input mode. In the other input mode, each training or test sample consists of three equally spaced images in the azimuth, with an interval of  $120^\circ$ . The three indicators obtained for each car are shown in Table 2. Except for the Ford Taurus Wagon, the IoU of each car is higher than 0.9, the CD is lower than 0.071, and the NC is higher than 0.088, regardless of the input modes. The reason is that the details on the surface of the Ford Taurus Wagon model are more complex than those of other car models. When the number of azimuthal images in one sample changes from 1 to 3, the relative growth rates for mIoU and mNC are 2.53% and 0.65%, respectively, and the relative reduction rate for mCD is 21.44%. Using three azimuthal images as the input yields better metrics than one azimuthal image because significantly more structural information is implied in three azimuthal images than in one.

**Table 2.** Results under different input modes.

Category	IoU $\uparrow$ ( $\times 10^{-2}$ )		CD $\downarrow$ ( $\times 10^{-2}$ )		NC $\uparrow$ ( $\times 10^{-1}$ )	
	1	3	1	3	1	3
	Azimuth	Azimuths	Azimuth	Azimuths	Azimuth	Azimuths
1	95.63	97.44	4.17	3.05	9.53	9.58
2	87.05	89.40	13.99	12.45	8.51	8.59
3	94.28	93.49	5.29	5.55	9.40	9.37
4	90.11	93.39	7.08	4.88	9.26	9.35
5	90.60	94.16	6.82	4.92	8.85	8.91
6	94.13	97.17	4.75	2.94	9.63	9.72
7	91.78	94.90	5.60	3.65	9.49	9.58
mean	91.94	94.27	6.81	5.35	9.24	9.30

Then, we also compare the proposed SFONet with other deep learning-based methods. When the input is one azimuthal image, Pixel2Mesh (P2M) [44], lightweight P2M (LP2M) [41], and ONet [45] are used for comparison. Specifically, the ONet's encoder is lightweight enough to match the small SAR datasets. When the input is three azimuthal images, R2N2 [42] and LRGT+ [62] are used for comparison. The values of mIoU, mCD, and mNC obtained by all these methods are shown in Table 3. Only the values of mIoU are calculated for R2N2 and LRGT+, since they are voxel-based representations. Moreover, the visualizations of the 3-D reconstruction results are all shown in Figure 8.



**Figure 8.** Visualization of seven cars. (a) Chevrolet Malibu. (b) Ford Taurus Wagon. (c) Nissan Sentra. (d) Toyota Camry. (e) Hyundai Santa Fe. (f) Nissan Maxima. (g) Chevrolet Prizm. The 3-D reconstruction methods used from top to bottom are P2M, LP2M, basic SFONet, R2N2, LRGT+, and SFONet with the pluggable module.

**Table 3.** Results of different deep learning-based methods.

Methods	mIoU $\uparrow$ ( $\times 10^{-2}$ )	mCD $\downarrow$ ( $\times 10^{-2}$ )	mNC $\uparrow$ ( $\times 10^{-1}$ )
P2M (1 azimuth)	68.17	18.39	8.35
LP2M (1 azimuth)	78.56	8.52	8.89
ONet (1 azimuth)	90.10	7.48	9.16
SFONet (1 azimuth)	<b>91.94</b>	<b>6.81</b>	<b>9.24</b>
R2N2 (3 azimuths)	75.14	-	-
LRGT+ (3 azimuths)	75.27	-	-
SFONet (3 azimuths)	<b>94.27</b>	<b>5.35</b>	<b>9.30</b>

Table 3 shows that the proposed SFONet achieves better metrics than those of the other methods when using one or three azimuthal images as the input. We explain the reasons by combining the visualization results. In Figure 8, the P2M obtains coarse 3D reconstruction results, with some holes on the surfaces of cars. Although the LP2M can eliminate this phenomenon, it lacks details for each car's surface. These two methods are based on the deformation of a random ellipsoid, so it is challenging for them to obtain complex topological structures. The ONet can obtain some details, such as rearview mirrors (in the red dashed ellipse) and wheels with textures (in the blue dashed ellipse), but the reconstructed surfaces are not smooth enough. However, the basic SFONet, using one azimuthal image as the input, can improve the smoothness of surfaces, such as the backs of cars (in the green dashed ellipse). The reason is that the SFONet uses a CV encoder, which can extract the target information implied in the phase data. Furthermore, the target shapes reconstructed by R2N2 and LRGT+ are correct, but these two voxel-based representations exhibit poor visualization. The SFONet with the pluggable module, using three azimuthal images as the input, can reconstruct smooth surfaces and provide sufficient details, such as the roof of a Ford Taurus Wagon and the rears of some other cars (in the purple dashed ellipse). In summary, the SFONet offers significant advantages in regards to detail reconstruction.

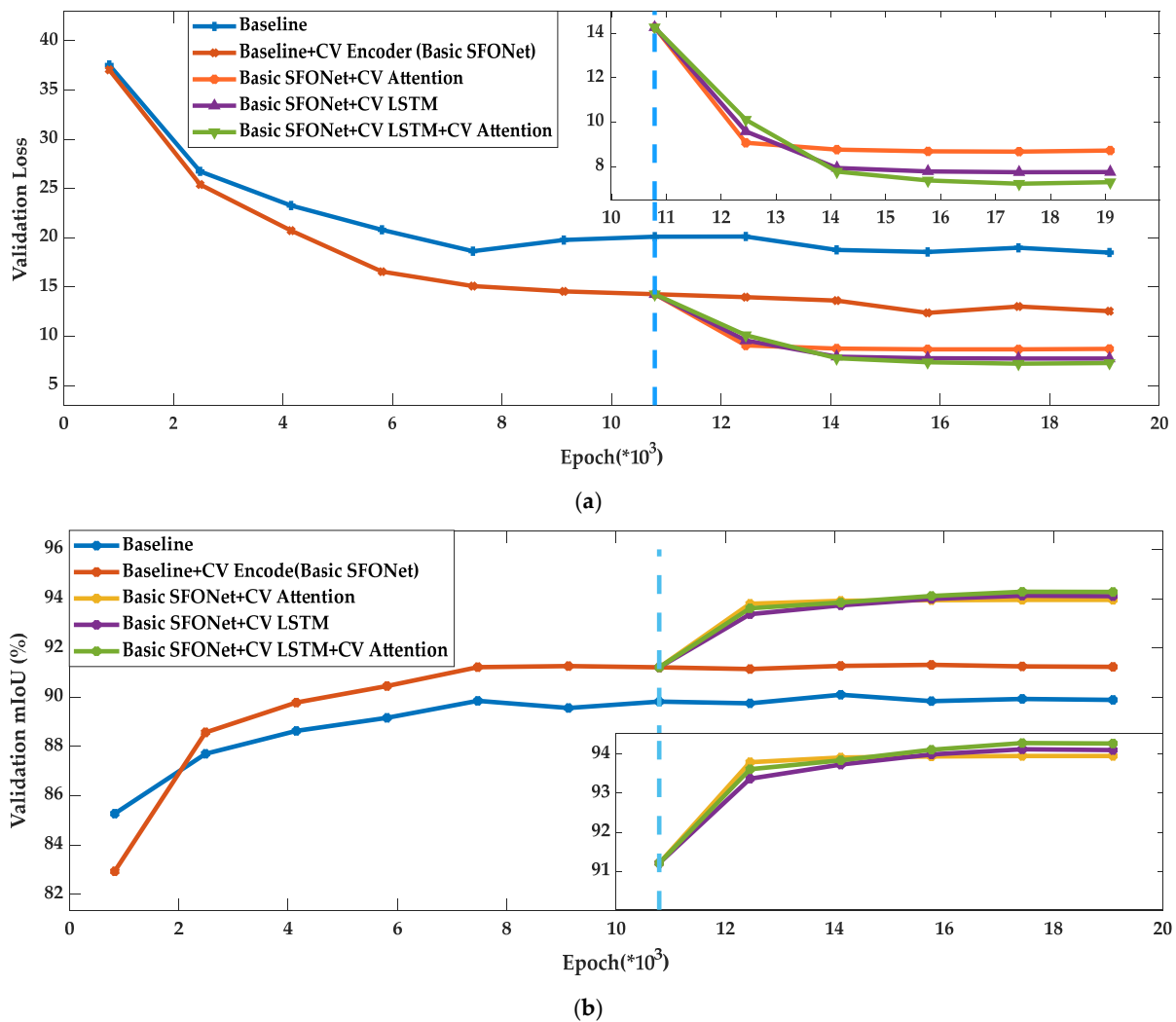
#### 4.4. Ablation Experiments

The lightweight ONet is used as the baseline. Then, the RV encoder is extended to a CV type to obtain the basic SFONet. The baseline and the basic SFONet use one azimuthal image as the input. Subsequently, we add the CV LSTM, CV attention, and both the CV LSTM and CV attention to the network. These three cases use three azimuthal images as their inputs. In Table 4, we list three metrics for five cases. The results indicate that the CV encoder can perform better than the RV type because the phase data of the SAR images also contain rich target information. Moreover, the CV LSTM or CV attention can improve all three metrics. This is because the CV LSTM can extract structural features, and the CV attention module can fuse these features (discussed in Section 5.1). As expected, when both the CV LSTM and CV attention modules are added into the network simultaneously, the highest mIoU and the lowest mCD are achieved.

**Table 4.** Results of ablation experiments.

Baseline	CV Encoder	CV LSTM	CV Attention	mIoU $\uparrow$ ( $\times 10^{-2}$ )	mCD $\downarrow$ ( $\times 10^{-2}$ )	mNC $\uparrow$ ( $\times 10^{-1}$ )
✓				90.10	7.48	9.16
✓	✓			91.94	6.81	9.24
✓	✓	✓		94.07	5.54	<b>9.30</b>
✓	✓		✓	93.95	5.61	<b>9.30</b>
✓	✓	✓	✓	<b>94.27</b>	<b>5.35</b>	<b>9.30</b>

The validation loss curves for five cases are shown in Figure 9a, and the validation mIoU curves are shown in Figure 9b. All these curves can converge. The blue vertical dashed line in the middle of these curves divides the validation process into two stages: the first stage on the left and the second on the right. For the three cases using three azimuthal images as their inputs, the validation loss or mIoU curves in the first stage are the same as those in the basic SFONet due to the two-stage training strategy. The orders of the convergence values of all the validation loss or mIoU curves are consistent with the reconstruction performance reflected by the three metrics in Table 4.

**Figure 9.** Ablation experiments. (a) Validation loss curves. (b) Validation mIoU curves.



#### 4.5. The Number of CV LSTM Layers

The number of CV LSTM layers is selected mainly based on computing resource consumption and reconstruction performance. As the number of layers increases, the computational load increases. In addition, when the numbers of layers in CV LSTM are 1, 2, and 3, respectively, three metrics are given in Table 5. The results indicate that all three metrics reach their optimal values when the number of layers is 2. Therefore, we set the layers of CV LSTM to 2 in previous experiments.

**Table 5.** Results of different CV LSTM layers.

Number of CV LSTM Layers	mIoU $\uparrow$ ( $\times 10^{-2}$ )	mCD $\downarrow$ ( $\times 10^{-2}$ )	mNC $\uparrow$ ( $\times 10^{-1}$ )
1	93.51	5.74	9.27
2	<b>94.27</b>	<b>5.35</b>	<b>9.30</b>
3	94.22	5.52	<b>9.30</b>

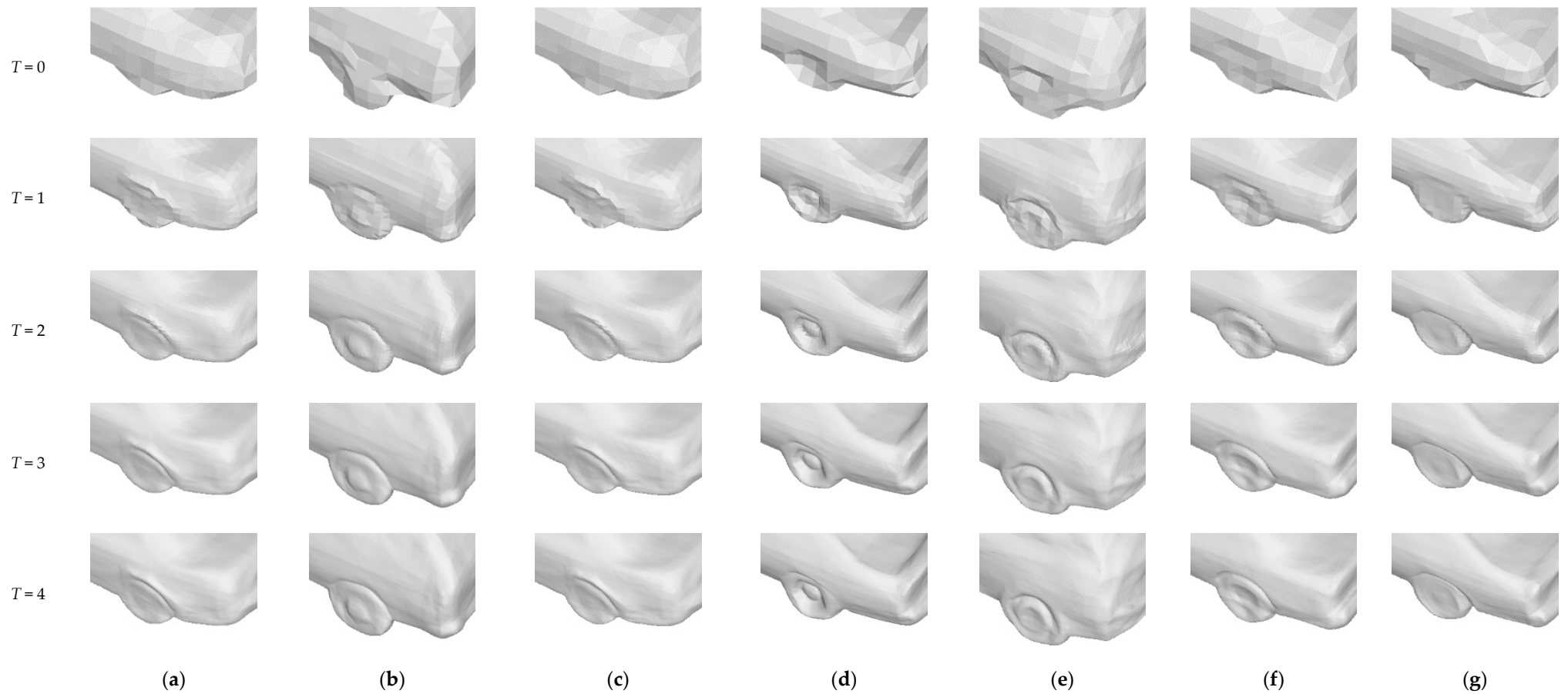
#### 4.6. The Number of Refinement Iterations in the Reference

The SFONet can obtain reconstruction results of any resolution through refinement in the reference process. However, the parameters to be calculated increase sharply with the increase in refinement iterations. Taking the Camry as an example, the numbers of refinement iterations T are 0, 1, 2, 3, and 4, respectively, and then the three metrics, the elapsed time, and the parameters are shown in Table 6. When T increases from 0 to 2, the improvement in the three metrics is significant. Afterwards, the performance improvement is relatively small when T increases from 2 to 4. On the other hand, as T increases, the number of grids increases exponentially, leading to a sharp increase in computation time and parameters. Therefore, we set the refinement iterations T to 2 in previous experiments to balance the reconstruction quality and computing resource consumption.

**Table 6.** Results of different refinement iterations.

Number of Refinement Iterations	mIoU $\uparrow$ ( $\times 10^{-2}$ )	mCD $\downarrow$ ( $\times 10^{-2}$ )	mNC $\uparrow$ ( $\times 10^{-1}$ )	Elapsed Time	Parameters
0	90.34	7.93	9.0	1.84 s	81 KB
1	93.59	5.84	9.23	1.92 s	371 KB
2	94.27	5.35	9.30	2.22 s	1602 KB
3	94.36	5.25	9.31	3.97 s	6638 KB
4	94.36	5.23	9.31	15.02 s	28,252 KB

In Figure 10, we also provide visualization results of a wheel and partial surface when T increases from 0 to 4. If the results are not refined (T = 0), the reconstruction results of seven cars are poor. With the increase in T, the wheel's texture becomes clearer and the surface smoother.



**Figure 10.** Partial visualization of reconstruction results at different refinement levels. (a) Chevrolet Malibu. (b) Ford Taurus Wagon. (c) Nissan Sentra. (d) Toyota Camry. (e) Hyundai Santa Fe. (f) Nissan Maxima. (g) Chevrolet Prizm.

## 5. Discussion

We discuss the roles of two submodules: CV LSTM and CV attention. We also analyze the influences of various compositions of training samples, including the azimuthal interval, the number of images, the number of passes, and the number of sub-apertures per pass, and then discuss the reasonable composition of one training sample.

### 5.1. Roles of CV LSTM and CV Attention

The cosine similarity is used to discuss the roles of CV LSTM and CV attention. The cosine similarity between two 1-D vectors  $\mathbf{a}$  and  $\mathbf{b}$  can be defined as

$$S = \frac{\mathbf{a} \bullet \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} \quad (15)$$

where  $\bullet$  denotes the dot-product, and  $\|\cdot\|$  denotes the modulus of a vector. For two image matrices, their cosine similarity can also be calculated by (15) after flattening them into two 1-D vectors.

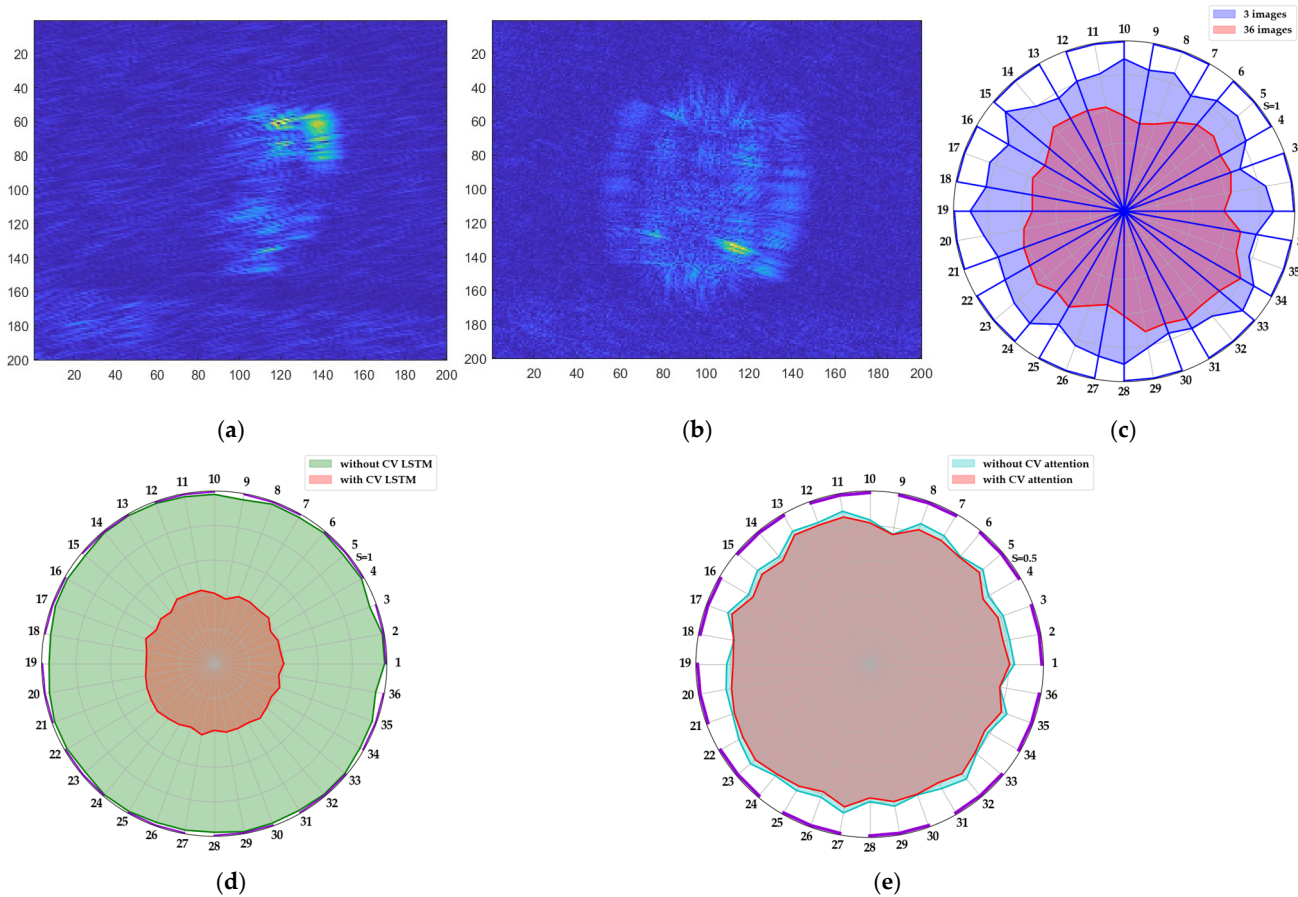
We obtain 36 azimuthal images of the Camry using HH polarimetric SAR data from the fifth pass and calculate the cosine similarity values between the 2-D SAR images in two cases. The first case is used to divide 36 images into 12 test samples, and each test sample includes three adjacent azimuthal images. For each test sample, we calculate the cosine similarity value between each azimuthal amplitude image  $I_i$  ( $i = 1, 2, 3$ ) and the synthetic amplitude image  $I$  after the coherent accumulation of three images. The synthetic amplitude image of the first test sample is shown in Figure 11a. All the cosine similarity values of 12 test samples are shown in Figure 11c (in blue). The second case is used to calculate the cosine similarity value between each azimuthal image  $I_i$  ( $i = 1, 2, \dots, 36$ ) and the synthetic amplitude image  $I'$  after the coherent accumulation of 36 images. The image  $I'$  is shown in Figure 11b. All the cosine similarity values are shown in Figure 11c (in red).

Comparing Figure 11a with Figure 11b, there is more deterministic structural information about the target in Figure 11b than in Figure 11a. Correspondingly, the similarity values (in red) are smaller than those (in blue) in Figure 11c. This indicates that the cosine similarity value between one azimuthal image, with a certain degree of randomness, and the synthetic image decreases with the increasing deterministic structural information implied in the synthetic image.

Then, we illustrate the role of CV LSTM by using the above conclusion. The above 36 azimuthal images of the Camry are still used. Suppose that each input of the SFONet is a test sample consisting of three azimuthal images. For each image  $I_i$  ( $i = 1, 2, 3$ ), the corresponding feature vector  $F_i$  can be obtained after the CV encoding. When the CV LSTM is added into the SFONet, the fusion feature after CV LSTM and CV attention is denoted as  $F$ . We calculate the cosine similarity value between each feature vector  $F_i$  ( $i = 1, 2, 3$ ) and  $F$  for each test sample. All the cosine similarity values of 12 test samples are shown in Figure 11d (in red). When the CV LSTM is not added to the SFONet, the fusion feature vector after the CV attention is denoted as  $F'$ . We also calculate the cosine similarity value between each feature vector  $F_i$  ( $i = 1, 2, 3$ ) and  $F'$  for each test sample. All the cosine similarity values of 12 test samples are shown in Figure 11d (in green). From Figure 11d, we can deduce that the CV LSTM can obtain rich structural information about the target, since the cosine similarity values in red are much smaller than those in green.

We illustrate the role of CV attention in the same way. When the CV attention is added to the network, all the cosine similarity values of 12 test samples are shown in Figure 11e (in red), which are the same as those shown in Figure 11d (in red). When the CV attention is not added to the network, the final output feature vector  $F''$  comes from the last time step of the CV LSTM. We calculate the cosine similarity values between the feature vectors

$F_i$  ( $i = 1, 2, 3$ ) and  $F''$  for each test sample, and all the cosine similarity values of 12 test samples are shown in Figure 11e (in cyan). From Figure 11e, we can deduce that the CV attention further obtains a small amount of structural information from other time steps of the CV LSTM, since the cosine similarity values in red are slightly smaller than those in cyan.



**Figure 11.** (a) Synthetic amplitude image after the coherent accumulation of three adjacent azimuthal images. (b) Synthetic amplitude image after the coherent accumulation of 36 azimuthal images. (c) Cosine similarity values in two cases. One is between each azimuthal image and the synthetic amplitude image of 3 images; the other is between each azimuthal image and the synthetic amplitude image of 36 images. (d) Cosine similarity values are obtained with/without CV LSTM (in red/green). (e) Cosine similarity values are obtained with/without CV attention (in red/cyan).

## 5.2. Influence of the Composition of Training Samples

When one training sample consists of multiple azimuthal images, both the azimuthal interval between two adjacent images and the number of images affect the reconstructed performance. In addition, the number of passes and training samples per pass can also affect the reconstruction performance.

### 5.2.1. Influence of the Azimuthal Interval

All the images obtained from the first three passes are used as training samples, and each contains three images with equal azimuthal intervals. The azimuth intervals between two adjacent images are taken as  $10^\circ$ ,  $30^\circ$ ,  $60^\circ$ ,  $90^\circ$ , and  $120^\circ$ , respectively, and then three mean metrics on the same test sample are listed in Table 7. The results show that the reconstruction performance improves as the azimuthal interval increases. When the azimuthal interval is changed from  $10^\circ$  to  $120^\circ$ , the relative growth rates for mIoU and mNC are 1.45% and 0.54%, respectively, and the relative reduction rate for mCD is 12.3%.

The reason is that when the azimuthal interval is small, information redundancy exists in these azimuthal features extracted from three images, and the overall information is limited. However, when the azimuthal interval is large, CV LSTM can extract rich structural features to improve 3-D reconstruction performance.

**Table 7.** Influence of the azimuthal interval.

Azimuthal Interval	mIoU $\uparrow$ ( $\times 10^{-2}$ )	mCD $\downarrow$ ( $\times 10^{-2}$ )	mNC $\uparrow$ ( $\times 10^{-1}$ )
10°	92.92	6.10	9.25
30°	93.32	5.85	9.26
60°	93.90	5.56	9.28
90°	94.10	5.41	9.29
120°	<b>94.27</b>	<b>5.35</b>	<b>9.30</b>

### 5.2.2. Influence of the Number of Images

The training samples are obtained from the first three passes, and each training sample contains multiple images with an equal azimuthal interval of 10°. The numbers of images are taken as 1, 2, 3, and 4, respectively, and then three metrics for the same test sample are listed in Table 8. The performance improves as the number of input images increases. When the number is changed from 2 to 4, the relative growth rates on mIoU and mNC are 1.36% and 0.22%, respectively, and the relative reduction rate using mCD is 12.9%. This is because when the azimuthal interval is fixed, there are more images, and richer structural information can be extracted.

**Table 8.** Influence of the number of images.

Number of Images	mIoU $\uparrow$ ( $\times 10^{-2}$ )	mCD $\downarrow$ ( $\times 10^{-2}$ )	mNC $\uparrow$ ( $\times 10^{-1}$ )
1	91.94	6.81	9.24
2	92.78	6.26	9.24
3	92.92	6.10	9.25
4	<b>93.19</b>	<b>5.93</b>	<b>9.26</b>

### 5.2.3. Influence of the Number of Passes

Each training sample is fixed to contain three images with an equal azimuthal interval of 120°. The numbers of passes are taken as 1, 2, and 3, respectively, and then three metrics for the same test sample are listed in Table 9. Although the reconstruction performance gradually improves as the number of passes increases, this improvement is insignificant. When the number is changed from 1 to 3, the relative growth rates of mIoU and mNC are 0.37% and 0.54%, respectively, and the relative reduction rate of mCD is 3.78%. The reason is that when test samples in one pass can obtain sufficient information about the target, much of the information from the other passes will become redundant. At this point, the number of passes required for the network training can be reduced.

**Table 9.** Influence of the number of passes.

Number of Passes	mIoU $\uparrow$ ( $\times 10^{-2}$ )	mCD $\downarrow$ ( $\times 10^{-2}$ )	mNC $\uparrow$ ( $\times 10^{-1}$ )
1	93.92	5.56	9.25
2	94.08	5.47	9.27
3	<b>94.27</b>	<b>5.35</b>	<b>9.30</b>

#### 5.2.4. Influence of the Number of Sub-Apertures per Pass

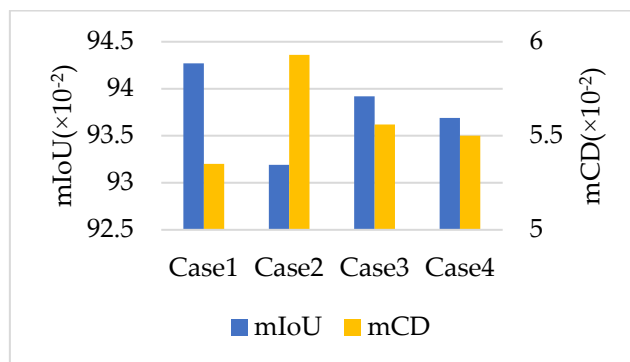
The images from the first three passes are used as training samples. For each pass, we uniformly downsample 36 sub-apertures into 6, 12, 18, and 36 sub-apertures, respectively, and randomly choose the starting sub-aperture number. Then, we choose three images with an equal azimuthal interval of  $120^\circ$  from these downsampled sub-aperture sequences as one training sample. Obviously, under equal azimuthal intervals, the more sub-apertures there are, the more training samples can be obtained. Table 10 lists three metrics for the same test sample under different downsampling cases. The results show that the number of sub-apertures per pass severely affects the reconstruction performance. When the number is changed from 36 to 6, the relative reduction rates for mIoU and mNC are 1.52% and 0.86%, respectively, and the relative growth rate for mCD is 26.54%. This is because fewer samples per pass provide less structural information for the target, leading to worse reconstruction performance.

**Table 10.** Influence of the number of sub-apertures per pass.

Number of Sub-Apertures per Pass	mIoU $\uparrow$ ( $\times 10^{-2}$ )	mCD $\downarrow$ ( $\times 10^{-2}$ )	mNC $\uparrow$ ( $\times 10^{-1}$ )
6	92.84	6.77	9.22
12	93.69	5.50	9.28
18	94.16	5.79	<b>9.30</b>
36	<b>94.27</b>	<b>5.35</b>	<b>9.30</b>

#### 5.2.5. Discussion About the Composition of One Training Sample

For the convenience of discussion, we define four cases. Case 1: Training samples are from the first three passes, and each sample contains three azimuthal images with an equal azimuthal interval of  $120^\circ$ . Case 2: Training samples are from the first three passes, and each sample contains four azimuthal images with an equal azimuthal interval of  $10^\circ$ . Case 3: Training samples are from the first pass, and each sample contains three azimuthal images with an equal azimuthal interval of  $120^\circ$ . Case 4: Training samples are from the first three passes; the number of sub-apertures per pass is 12; each sample contains three azimuthal images with an equal azimuthal interval of  $120^\circ$ . Three metrics of the four cases are shown in Tables 7–10, respectively, where the mIoUs and mCDs of four cases are visualized in Figure 12. Comparing the metrics of Case 1 with those of Case 2, the performance of the former is superior to that of the latter. This means that even if there are fewer input images in a sample, the reconstruction performance may still be better due to the larger azimuthal interval. Comparing the metrics of Case 2 with those of Case 3, the mIoU and mCD of the former are inferior to those of the latter. This means that sufficient azimuthal information from a training sample helps reduce the number of passes, thereby reducing the total number of training samples. Comparing the metrics of Case 2 with those of Case 4, the performance of the former is inferior to that of the latter. It also indicates that a reasonable composition of one training sample can help reduce the number of training samples.



**Figure 12.** Comparison of mIoUs and mCDs for four cases.

## 6. Conclusions

In this paper, we propose a SAR-tailored SFONet for reconstructed 3-D targets using one or multiple 2-D images. It includes a basic network and one pluggable module. The basic network can reconstruct a 3-D target from one azimuthal image. When the pluggable module is added to the basic network, a refined 3-D target can be reconstructed from multiple azimuthal images. The lightweight CV encoder module in the basic network extracts features from CV SAR images, fully utilizing the characteristics of the SAR data. The pluggable module, including CV LSTM and attention submodules, can extract and fuse the structural features of the target from multiple azimuthal images. Furthermore, when both the single-input and multiple-input modes of the SFONet coexist, we also propose a two-stage training strategy to enhance the network performance, while improving training efficiency. In the first stage, the basic network is trained. There is no blurring, since one azimuthal image corresponds to one 3-D ground truth. In the second stage, only the pluggable module is trained. It can focus on extracting the structural information of the target from multiple azimuthal images. Once the SFONet is trained, a small amount of echo data must be collected to obtain 2-D images for the 3-D target reconstruction, reducing the hardware burden of the data acquisition. Finally, we construct an experimental dataset containing 2-D images and 3-D ground truth utilizing the Gotcha echo dataset. The main experimental conclusions are as follows: (1) The SFONet performs better than other deep learning-based methods using one or multiple SAR images as the input. (2) The lightweight CV encoder and the pluggable module effectively improve the 3-D reconstruction performance. (3) The number of refinement iterations in the inference affects reconstruction quality and computing resource consumption, requiring a compromising choice. (4) A reasonable composition of a training sample helps reduce the number of training samples. However, the SFONet still suffers from overfitting when the number of training samples is very small. In the future, we will investigate the 3-D reconstruction using few-shot samples.

**Author Contributions:** Conceptualization, L.Y. and W.H.; methodology, X.X.; software, J.L.; validation, M.L. and X.Y.; resources, H.B.; writing, L.Y. and X.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China (no. 62261027 and no. 62266020), the Natural Science Foundation of Jiangxi Province (no. 20224BAB202002, no. 20224BAB212008, and no. 20224BAB212013), the Science and Technology Project of Jiangxi Provincial Education Department (no. GJJ211410), and the Jiangxi Provincial Key Laboratory of Multidimensional Intelligent Perception and Control of China (no. 2024SSY03161).

**Data Availability Statement:** The Gotcha echo data is available at <https://www.sdms.afrl.af.mil/index.php?collection=gotcha>, accessed on 8 June 2024. The 3-D CAD models of seven cars are available at <https://3dwarehouse.sketchup.com/search/models>, accessed on 8 June 2024.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Zhu, X.X.; Bamler, R. Superresolving SAR tomography for multidimensional imaging of urban areas: Compressive sensing-based TomoSAR inversion. *IEEE Signal Process. Mag.* **2014**, *31*, 51–58. [\[CrossRef\]](#)
2. Zhang, H.; Lin, Y.; Teng, F.; Feng, S.; Yang, B.; Hong, W. Circular SAR incoherent 3D imaging with a NeRF-Inspired method. *Remote Sens.* **2023**, *15*, 3322. [\[CrossRef\]](#)
3. Lei, Z.; Xu, F.; Wei, J.; Cai, F.; Wang, F.; Jin, Y.-Q. SAR-NeRF: Neural radiance fields for synthetic aperture radar multiview representation. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5221415. [\[CrossRef\]](#)
4. Liu, A.; Zhang, S.; Zhang, C.; Zhi, S.; Li, X. RaNeRF: Neural 3-D reconstruction of space targets from ISAR image sequences. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5107215. [\[CrossRef\]](#)
5. Reigber, A.; Moreira, A. First demonstration of airborne SAR tomography using multibaseline L-band data. *IEEE Trans. Geosci. Remote Sens.* **2000**, *38*, 2142–2152. [\[CrossRef\]](#)
6. Fornaro, G.; Lombardini, F.; Serafino, F. Three-dimensional multipass SAR focusing: Experiments with long-term spaceborne data. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 702–714. [\[CrossRef\]](#)
7. Zhu, X.X.; Bamler, R. Very high resolution spaceborne SAR tomography in urban environment. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 4296–4308. [\[CrossRef\]](#)
8. Wang, Z.; Ding, Z.; Sun, T.; Zhao, J.; Wang, Y.; Zeng, T. UAV-based P-band SAR tomography with long baseline: A multimaster approach. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5207221. [\[CrossRef\]](#)
9. Liu, M.; Wang, Y.; Ding, Z.; Li, L.; Zeng, T. Atomic norm minimization based fast off-grid tomographic SAR imaging with nonuniform sampling. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5203517. [\[CrossRef\]](#)
10. Lombardini, F.; Montanari, M.; Gini, F. Reflectivity estimation for multibaseline interferometric radar imaging of layover extended sources. *IEEE Trans. Signal Process.* **2003**, *51*, 1508–1519. [\[CrossRef\]](#)
11. Shi, Y.; Zhu, X.X.; Yin, W.; Bamler, R. A fast and accurate basis pursuit denoising algorithm with application to super-resolving tomographic SAR. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6148–6158. [\[CrossRef\]](#)
12. Ponce, O.; Prats-Iraola, P.; Scheiber, R.; Reigber, A.; Moreira, A. First airborne demonstration of holographic SAR tomography with fully polarimetric multicircular acquisitions at L-band. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6170–6196. [\[CrossRef\]](#)
13. Bao, Q.; Lin, Y.; Hong, W.; Shen, W.; Zhao, Y.; Peng, X. Holographic SAR tomography image reconstruction by combination of adaptive imaging and sparse bayesian inference. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1248–1252. [\[CrossRef\]](#)
14. Wu, K.; Shen, Q.; Cui, W. 3-D tomographic circular SAR imaging of targets using scattering phase correction. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5221914. [\[CrossRef\]](#)
15. Bi, H.; Feng, J.; Jin, S.; Yang, W.; Xu, W. Mixed-Norm regularization-based polarimetric holographic SAR 3-D imaging. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 4002305. [\[CrossRef\]](#)
16. Rambour, C.; Denis, L.; Tupin, F.; Oriot, H.M. Introducing spatial regularization in SAR tomography reconstruction. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 8600–8617. [\[CrossRef\]](#)
17. Wang, X.; Dong, Z.; Wang, Y.; Chen, X.; Yu, A. Three-dimensional reconstruction of partially coherent scatterers using iterative sub-network generation method. *Remote Sens.* **2024**, *16*, 3707. [\[CrossRef\]](#)
18. Tebaldini, S.; Rocca, F. Multibaseline polarimetric SAR tomography of a boreal forest at P-and L-bands. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 232–246. [\[CrossRef\]](#)
19. Ngo, Y.-N.; Minh, D.H.T.; Baghdadi, N.; Fayad, I.; Ferro-Famil, L.; Huang, Y. Exploring tropical forests with GEDI and 3D SAR tomography. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 2503605. [\[CrossRef\]](#)
20. Jin, S.; Bi, H.; Guo, Q.; Zhang, J.; Hong, W. Iterative adaptive based multi-polarimetric SAR tomography of the forested areas. *Remote Sens.* **2024**, *16*, 1605. [\[CrossRef\]](#)
21. Tebaldini, S.; Rocca, F.; Meta, A.; Coccia, A. 3D imaging of an alpine glacier: Signal processing of data from the AlpTomoSAR campaign. In Proceedings of the European Radar Conference (EuRAD), Paris, France, 9–11 September 2015; pp. 37–40.
22. Nouvel, J.; Jeuland, H.; Bonin, G.; Roques, S.; Du Plessis, O.; Peyret, J. A Ka band imaging radar: DRIVE on board ONERA motorglider. In Proceedings of the IEEE International Symposium on Geoscience and Remote Sensing, Denver, CO, USA, 31 July–4 August 2006; pp. 134–136.
23. Weiss, M.; Gilles, M. Initial ARTINO radar experiments. In Proceedings of the 8th European Conference on Synthetic Aperture Radar, Aachen, Germany, 7–10 June 2010; pp. 1–4.
24. Peng, X.; Tan, W.; Hong, W.; Jiang, C.; Bao, Q.; Wang, Y. Airborne DLSLA 3-D SAR image reconstruction by combination of polar formatting and L1 regularization. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 213–226. [\[CrossRef\]](#)
25. Qiu, X.; Luo, Y.; Song, S.; Peng, L.; Cheng, Y.; Yan, Q.; ShangGuan, S.; Jiao, Z.; Zhang, Z.; Ding, C. Microwave vision three-dimensional SAR experimental system and full-polarimetric data processing method. *J. Radars* **2024**, *13*, 941–954. [\[CrossRef\]](#)



26. Zhang, F.; Liang, X.; Wu, Y.; Lv, X. 3D surface reconstruction of layover areas in continuous terrain for multi-baseline SAR interferometry using a curve model. *Int. J. Remote Sens.* **2015**, *36*, 2093–2112. [[CrossRef](#)]
27. Wang, J.; Chen, L.-Y.; Liang, X.-D.; Ding, C.-b.; Li, K. Implementation of the OFDM chirp waveform on MIMO SAR systems. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 5218–5228. [[CrossRef](#)]
28. Hu, F.; Wang, F.; Yu, H.; Xu, F. Asymptotic 3-D phase unwrapping for very sparse airborne array InSAR images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5234115. [[CrossRef](#)]
29. LI, H.; LIANG, X.; ZHANG, F.; Ding, C.; Wu, Y. A novel 3-D reconstruction approach based on group sparsity of array InSAR. *Sci. Sin. Inform.* **2018**, *48*, 1051–1064. [[CrossRef](#)]
30. Jiao, Z.; Ding, C.; Qiu, X.; Zhou, L.; Chen, L.; Han, D.; Guo, J. Urban 3D imaging using airborne TomoSAR: Contextual information-based approach in the statistical way. *ISPRS J. Photogramm. Remote Sens.* **2020**, *170*, 127–141. [[CrossRef](#)]
31. Cui, C.; Liu, Y.; Zhang, F.; Shi, M.; Chen, L.; Li, W.; Li, Z. A novel automatic registration method for array InSAR point clouds in urban scenes. *Remote Sens.* **2024**, *16*, 601. [[CrossRef](#)]
32. Li, Z.; Zhang, F.; Wan, Y.; Chen, L.; Wang, D.; Yang, L. Airborne circular flight array SAR 3-D imaging algorithm of buildings based on layered phase compensation in the wavenumber domain. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5213512. [[CrossRef](#)]
33. Wang, S.; Guo, J.; Zhang, Y.; Wu, Y. Multi-baseline SAR 3D reconstruction of vehicle from very sparse aspects: A generative adversarial network based approach. *ISPRS J. Photogramm. Remote Sens.* **2023**, *197*, 36–55. [[CrossRef](#)]
34. Budillon, A.; Johnsny, A.C.; Schirinzi, G.; Vitale, S. SAR tomography based on deep learning. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 3625–3628.
35. Wang, M.; Zhang, Z.; Qiu, X.; Gao, S.; Wang, Y. ATASI-Net: An efficient sparse reconstruction network for tomographic SAR imaging with adaptive threshold. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 4701918. [[CrossRef](#)]
36. Qian, K.; Wang, Y.; Jung, P.; Shi, Y.; Zhu, X.X. Basis pursuit denoising via recurrent neural network applied to super-resolving SAR tomography. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 4710015. [[CrossRef](#)]
37. Wang, M.; Wei, S.; Zhou, Z.; Shi, J.; Zhang, X.; Guo, Y. CTV-Net: Complex-valued TV-driven network with nested topology for 3-D SAR imaging. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 5588–5602. [[CrossRef](#)] [[PubMed](#)]
38. Wang, Y.; Liu, C.; Zhu, R.; Liu, M.; Ding, Z.; Zeng, T. MAda-Net: Model-adaptive deep learning imaging for SAR tomography. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5202413. [[CrossRef](#)]
39. Chen, J.; Peng, L.; Qiu, X.; Ding, C.; Wu, Y. A 3D building reconstruction method for SAR images based on deep neural network. *Sci. Sin. Inform.* **2019**, *49*, 1606–1625. [[CrossRef](#)]
40. Yang, Z.-L.; Zhou, R.-Y.; Wang, F.; Xu, F. A point clouds framework for 3-D reconstruction of SAR images based on 3-D parametric electromagnetic part model. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Brussels, Belgium, 11–16 July 2021; pp. 4818–4821.
41. Yu, L.; Zou, J.; Liang, M.; Li, L.; Xie, X.; Yu, X.; Hong, W. Lightweight pixel2mesh for 3-D target reconstruction from a single SAR image. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 4000805. [[CrossRef](#)]
42. Choy, C.B.; Xu, D.; Gwak, J.; Chen, K.; Savarese, S. 3D-R2N2: A unified approach for single and multi-view 3-D object reconstruction. In Proceedings of the 2016 European Conference on Computer Vision (ECCV), Berlin, Germany, 11–14 October 2016; pp. 628–644.
43. Achlioptas, P.; Diamanti, O.; Mitliagkas, I.; Guibas, L. Learning representations and generative models for 3-D point clouds. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 40–49.
44. Wang, N.; Zhang, Y.; Li, Z.; Fu, Y.; Liu, W.; Jiang, Y.-G. Pixel2mesh: Generating 3-D mesh models from single RGB images. In Proceedings of the 2018 European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 52–67.
45. Mescheder, L.; Oechsle, M.; Niemeyer, M.; Nowozin, S.; Geiger, A. Occupancy networks: Learning 3-D reconstruction in function space. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4460–4470.
46. Chen, Z.; Zhang, H. Learning implicit fields for generative shape modeling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 5932–5941.
47. Tatarchenko, M.; Dosovitskiy, A.; Brox, T. Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2088–2096.
48. Lorensen, W.E.; Cline, H.E. Marching cubes: A high resolution 3D surface construction algorithm. *ACM Siggraph Comput. Graph.* **1987**, *21*, 163–169. [[CrossRef](#)]
49. Peng, S.; Niemeyer, M.; Mescheder, L.; Pollefeys, M.; Geiger, A. Convolutional occupancy networks. In Proceedings of the 2020 European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 523–540.
50. Lionar, S.; Emtsev, D.; Svilarkovic, D.; Peng, S. Dynamic plane convolutional occupancy networks. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Virtual, 5–9 January 2021; pp. 1829–1838.

51. Zhao, C.; Zhang, C.; Yan, Y.; Su, N. A 3D reconstruction framework of buildings using single off-nadir satellite image. *Remote Sens.* **2021**, *13*, 4434. [[CrossRef](#)]
52. Schmidhuber, J.; Hochreiter, S. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
53. Lattari, F.; Rucci, A.; Matteucci, M. A deep learning approach for change points detection in InSAR time series. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5223916. [[CrossRef](#)]
54. Kulshrestha, A.; Chang, L.; Stein, A. Use of LSTM for sinkhole-related anomaly detection and classification of InSAR deformation time series. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2022**, *15*, 4559–4570. [[CrossRef](#)]
55. Ding, J.; Wen, L.; Zhong, C.; Loffeld, O. Video SAR moving target indication using deep neural network. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 7194–7204. [[CrossRef](#)]
56. Zhou, Y.; Shi, J.; Wang, C.; Hu, Y.; Zhou, Z.; Yang, X.; Zhang, X.; Wei, S. SAR ground moving target refocusing by combining mRe<sup>3</sup> network and TV $\beta$ -LSTM. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5200814. [[CrossRef](#)]
57. Wang, C.; Liu, X.; Pei, J.; Huang, Y.; Zhang, Y.; Yang, J. Multiview attention CNN-LSTM network for SAR automatic target recognition. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2021**, *14*, 12504–12513. [[CrossRef](#)]
58. Bai, X.; Xue, R.; Wang, L.; Zhou, F. Sequence SAR image classification based on bidirectional convolution-recurrent network. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9223–9235. [[CrossRef](#)]
59. Ni, J.; Zhang, F.; Yin, Q.; Zhou, Y.; Li, H.-C.; Hong, W. Random neighbor pixel-block-based deep recurrent learning for polarimetric SAR image classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 7557–7569. [[CrossRef](#)]
60. Yang, B.; Wang, S.; Markham, A.; Trigoni, N. Robust attentional aggregation of deep feature sets for multi-view 3D reconstruction. *Int. J. Comput. Vis.* **2019**, *128*, 53–73. [[CrossRef](#)]
61. Dungan, K.E.; Potter, L.C. Classifying vehicles in wide-angle radar using pyramid match hashing. *IEEE J. Sel. Topics Signal Process.* **2011**, *5*, 577–591. [[CrossRef](#)]
62. Yang, L.; Zhu, Z.; Lin, X.; Nong, J.; Liang, Y. Long-range grouping transformer for multi-view 3D reconstruction. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; pp. 18257–18267.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.