

# ES-MVSNet: Efficient Framework for End-to-end Self-supervised Multi-View Stereo

Qiang Zhou

mightyza@gmail.com

Chaohui Yu

huakun.ych@alibaba-inc.com

Jingliang Li

lijingliang20@mails.ucas.ac.cn

Yuang Liu

frankliu624@gmail.com

Jing Wang

yunfei.wj@alibaba-inc.com

Zhibin Wang

zhibin.waz@alibaba-inc.com

## Abstract

Compared to the multi-stage self-supervised multi-view stereo (MVS) method, the end-to-end (E2E) approach has received more attention due to its concise and efficient training pipeline. Recent E2E self-supervised MVS approaches have integrated third-party models (such as optical flow models, semantic segmentation models, NeRF models, etc.) to provide additional consistency constraints, which grows GPU memory consumption and complicates the model’s structure and training pipeline. In this work, we propose an efficient framework for end-to-end self-supervised MVS, dubbed ES-MVSNet. To alleviate the high memory consumption of current E2E self-supervised MVS frameworks, we present a memory-efficient architecture that reduces memory usage by 43% without compromising model performance. Furthermore, with the novel design of asymmetric view selection policy and region-aware depth consistency, we achieve state-of-the-art performance among E2E self-supervised MVS methods, without relying on third-party models for additional consistency signals. Extensive experiments on DTU and Tanks&Temples benchmarks demonstrate that the proposed ES-MVSNet approach achieves state-of-the-art performance among E2E self-supervised MVS methods and competitive performance to many supervised and multi-stage self-supervised methods.

## 1. Introduction

Multi-View Stereo (MVS) [23] is a long-standing fundamental task in 3D computer vision, aiming to recover 3D point clouds of real scenes from multi-view images and corresponding calibrated cameras. Like on other vision tasks, deep learning has driven the rapid development of MVS in recent years. Especially on public datasets like DTU [2] and Tanks&Temples [15], the end-to-end MVS depth estimation models [33, 34, 8, 26, 19] significantly improve the reconstruction performance compared with traditional

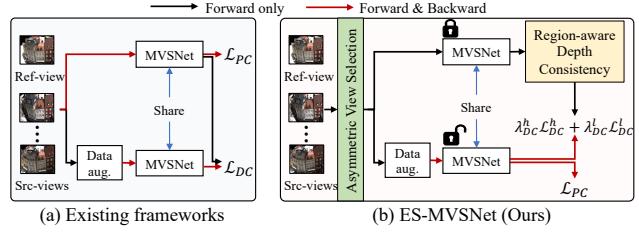


Figure 1: High-level comparison between ES-MVSNet and existing E2E self-supervised MVS frameworks [27, 3, 28].  $\mathcal{L}_{PC}$ ,  $\mathcal{L}_{DC}$  denote photometric and depth consistency losses, respectively.

geometry-based methods [21, 22]. MVSNet [33] is one of the representative MVS models based on fully supervised learning, which proposes to encode RGB information from different camera views into a cost volume, then predicts a depth map for point cloud reconstruction. The following supervised approaches [8, 34, 26, 17, 19] improve the neural network architecture, reduce memory usage, and acquire state-of-the-art depth estimation performance on multiple benchmarks. However, acquiring ground-truth depth data for supervision is difficult and expensive, limiting the practical use of these MVS models in general real-world scenarios. To reduce the reliance on ground-truth depth data, recently proposed self-supervised MVS methods [27, 3, 7] have received increasing attention.

Among the self-supervised MVS methods, the end-to-end (E2E) methods draw more attention due to their concise framework and efficient training. The commonly used framework is shown in Figure 1(a), in which the photometric consistency loss is applied in the weak-augmentation branch, and the depth consistency loss is applied in the strong-augmentation branch. The photometric consistency assumes that pixels belonging to the same 3D point have the same color properties in different views. The depth consistency leverages the predicted depth maps of the weak-augmentation branch to supervise the predictions of the strong-augmentation branch. To further improve perfor-

mance, recent works introduce other consistency signals from third-party models, such as optical flow, segmentation, and NeRF models [28, 27, 3]. The E2E self-supervised MVS framework consumes vast GPU memory due to the 3D-CNN for cost volume regularization. When additional third-party models are further introduced, the requirements for memory usage become more stringent, which significantly limits the application of these methods in high-resolution scenarios.

In this work, we propose an efficient E2E self-supervised MVS framework that does not introduce third-party models and achieves state-of-the-art performance on DTU [2] and Tanks&Temples [15] datasets. **Our first contribution** is to propose a memory-efficient structure, as shown in Figure 1(b). In our framework, photometric consistency and depth consistency losses are applied to the strong-augmentation branch, leaving the parameters of the weak-augmentation branch frozen. The weak-augmentation branch only needs to infer the pseudo-depth map without the backward operation required, which reduces the GPU memory usage by 43%. Furthermore, the model performance improves when the photometric consistency is applied to more diverse augmented samples. **Our second contribution** is to improve the efficacy of depth consistency. Specifically, we propose an asymmetric view selection policy. For the weak-augmentation branch, as in existing work, the top- $K$  source views are selected according to view-scores [33] to improve the accuracy of predicted pseudo-depth maps. For the strong-augmentation branch, we randomly select source views according to the view-scores to increase the view diversity between these two branches. Furthermore, we observe an apparent imbalance in the depth consistency loss among different regions, as shown in Figure 2, implying that using a single global loss weight is inadequate. We propose region-aware depth consistency to alleviate this issue, which partitions pseudo-depth maps into high-quality and low-quality regions via online cross-view checking and assigns them different loss weights. Figure 5 shows that the loss values of the two partitions differ by order of magnitude, verifying the necessity of our proposed region-aware loss weights and the effectiveness of the partitioning scheme based on online cross-view checking. The detailed structure of our framework is depicted in Figure 3.

Our method, dubbed ES-MVSNet, achieves state-of-the-art point cloud reconstruction results in the competitive DTU benchmark [2], and also demonstrates robust performance on out-of-distribution samples in the Tanks&Temples dataset [15]. In summary, our contributions are as follows:

- We propose an efficient framework for end-to-end self-supervised MVS, dubbed ES-MVSNet.
- We propose a memory-efficient architecture that re-

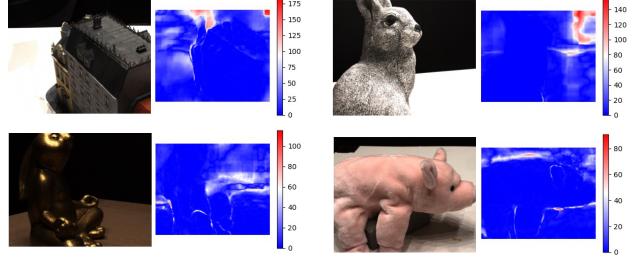


Figure 2: Imbalance in depth consistency loss. Some regions, such as the background and object boundaries, have a large loss value and may overwhelm others.

duces memory usage by 43% without compromising model performance, which benefits the use in high-resolution scenarios.

- We improve the efficacy of depth consistency through two novel designs: asymmetric view selection policy and region-aware depth consistency.

## 2. Related Work

### 2.1. Supervised MVS

Since the era of deep learning, many supervised learning-based models have been proposed to reconstruct 3D scenes [11, 12, 33, 8, 34, 26, 19, 17]. Among these methods, depth map reconstruction demonstrates the most versatility, as it decouples the intricate MVS problem into a per-view depth map estimation problem. One of the representative works is MVSNet [33], which encodes camera parameters and backbone features into a cost volume via homography warping, and regularizes the volume via a 3D CNN before predicting the final depth map. The 3D CNN in MVSNet consumes high GPU memory, which limits its usage in high-resolution scenes. To alleviate the memory usage problem, some works [26, 34, 30] propose replacing 3D CNNs with convolutional recurrent GRU [5] or LSTM [9] units. The subsequent multi-stage architectures [8, 32, 19] dramatically improve the model performance by learning depth predictions in a coarse-to-fine manner. The multi-stage architecture also balances model performance and memory cost better, where more depth hypotheses are employed in low-resolution stages and fewer in high-resolution stages.

However, the reliance on ground truth depth data limits the application of supervised MVS methods to a broader range of realistic scenarios. Therefore, it is essential to explore alternative self-supervised methods.

### 2.2. Multi-Stage Self-supervised MVS

Multi-stage self-supervised MVS usually first trains a teacher model based on photometric consistency. The teacher model is then used to generate pseudo-depth maps

to supervise the training of the student model. Several cycles may be iterated to improve the performance of the student model, i.e., the previous student model acts as a new teacher model to guide the learning of the new student model. For example, U-MVSNet [28] consists of two stages: self-supervised pre-training and pseudo-label-based post-training. In the first pre-training stage, a flow-depth consistency loss is introduced in addition to the photometric consistency loss. In the second post-training stage, the student model is supervised with pseudo-labels. CVP-MVSNet [32] proposes to learn an initial pseudo-depth map through unsupervised pre-training, then use a well-designed pipeline to refine the initial pseudo-depth map, and finally use the refined pseudo-depth map to supervise the training of the student model. The KD-MVS [7] first trains a teacher model in a self-supervised manner using photometric and feature consistency, then distills the knowledge from the teacher model to the student model via proposed probabilistic knowledge transfer.

Although multi-stage self-supervised MVS methods achieve superior performance, their complex framework and training process make it inconvenient to use in practical applications.

### 2.3. End-to-end Self-supervised MVS

Unlike the multi-stage approaches, the end-to-end approaches only train one model, making the training process more efficient and concise. Unsup\_MVS [14] proposes the first end-to-end self-supervised MVS framework, guiding the depth prediction by minimizing the discrepancy between the reference image and the inversely warped images from source views. The training objectives of photometric consistency, SSIM [25] loss, and depth smoothing used in this work are widely used in subsequent self-supervised work. JDACS [27] incorporates data augmentation into self-supervised MVS and applies depth consistency to constrain the prediction of weak and strong augmentation branches. It further proposes cross-view semantic consistency implemented via an unsupervised co-segmentation module. To alleviate the correspondence ambiguity among views due to occlusion, etc., RC-MVSNet [3] adds an additional NeRF [18] branch, which shares the backbone with the original MVS branch, and imposes rendering consistency on the depth maps predicted by these two branches.

In this work, instead of introducing additional consistency signals with third-party models such as segmentation and NeRF, we focus on improving the efficacy of depth consistency and propose a memory-efficient framework.

## 3. Method

In this section, we describe ES-MVSNet, our proposed approach for end-to-end self-supervised multi-view stereo. Given  $N$  images as input (by default, 1st is the reference

view), with their corresponding cameras' intrinsic and extrinsic parameters, our method predicts a depth map in the reference camera view. The overall pipeline is illustrated in Figure 3. In the following, we first describe the existing baseline framework in Section 3.1. Then we elaborate innovative designs of our approach, including memory-efficient design (Section 3.2), asymmetric view selection policy (Section 3.3), and region-aware depth consistency (Section 3.4).

### 3.1. Preliminary

This section introduces the model framework and training loss used in existing end-to-end self-supervised MVS work [27, 3, 28]. Third-party models such as optical flow, segmentation, and NeRF models are excluded for providing a clean baseline.

**Model Architecture.** The framework in Figure 1(a) is adopted by recent E2E self-supervised MVS methods, which contains two branches of weak-augmentation and strong-augmentation respectively. The two branch models have the same structure and share model parameters. The weak-augmentation branch takes the original image as input and finally applies a photometric consistency loss on the predicted depth map. The strong-augmentation branch applies color perturbation to the input image to simulate color inconsistency between views, and finally applies a depth consistency loss to the predicted depth map. For each branch, the network firstly extracts features using a CNN from  $N$  input images. Then a variance-based cost volume [33, 36] is constructed via differentiable homography warping and a 3D U-Net is used to regularize the 3D cost volume. Finally, the depth map is inferred for every reference image.

**Overall Loss.** The photometric consistency loss is applied to the weak-augmentation branch to minimize the difference between the warped image and the original image at the same view. For a particular image pair  $(I_1, I_j)$  with associated intrinsic and extrinsic parameters  $(K, T)$ , we can calculate the corresponding projected location  $p'_j$  in the source view from its coordinates  $p_1$  in the reference view:

$$p'_{j,k} = K_j T_{1j} (D_1(p_{1,k}) K_1^{-1} p_{1,k}), \quad (1)$$

where  $k(1 \leq k \leq HW)$  is the index of the pixels,  $H$  and  $W$  denote the images' height and width, and  $D_1$  represents the predicted depth map in the reference view. Through differentiable bilinear sampling, we can obtain the warped image  $I_1^j$  under the reference view, taking the source view image  $I_j$  and the predicted depth map  $D_1$  as input. Along with the warping, a binary validity mask  $M_j$  is generated simultaneously, indicating whether the projected position  $p'_j$

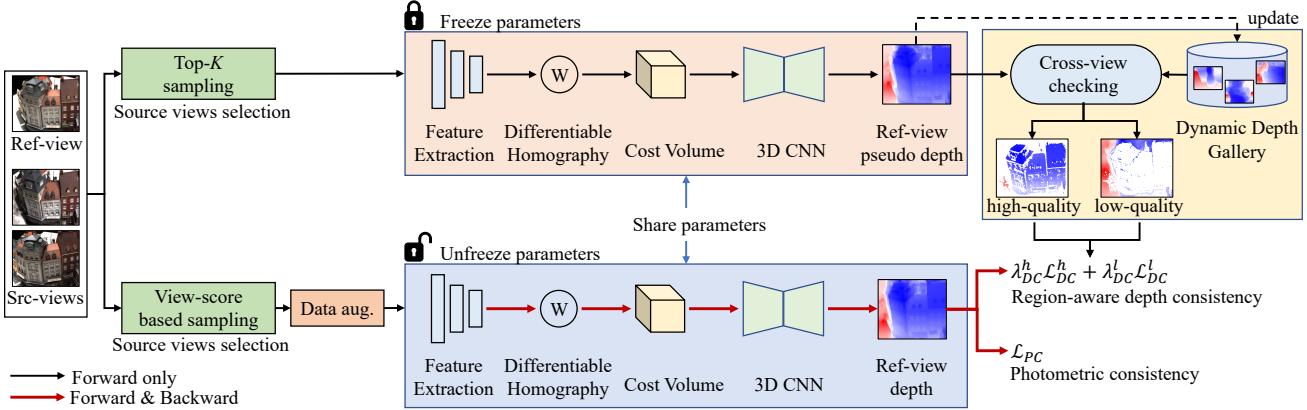


Figure 3: Overview of our proposed framework (ES-MVSNet) for end-to-end self-supervised MVS. Compared with previous methods, our proposed approach possesses three novel designs: memory-efficient design, asymmetric view selection strategy, and region-aware depth consistency.

lies within the valid image region. During training, all  $N - 1$  source views are warped to the reference view to compute the photometric loss:

$$\mathcal{L}_{PC} = \sum_{j=2}^N \frac{\|(I_1^j - I_1) \odot M_j\|_2 + \|(\nabla I_1^j - \nabla I_1) \odot M_j\|_2}{\|M_j\|_1}, \quad (2)$$

where  $\odot$  denotes Hadamard product,  $\nabla$  represents the gradient of pixels.

The depth consistency loss is applied to the predicted depth maps  $D_1^s$  of the strong-augmentation branch, taking the output depth maps  $D_1^w$  of the weak-augmentation branch as pseudo ground truths.

$$\mathcal{L}_{DC} = \frac{1}{HW} \sum_{k=1}^{HW} \|D_{1,k}^s - D_{1,k}^w\|_2. \quad (3)$$

The final training objective can be constructed as follows:

$$\begin{aligned} \mathcal{L} = & \lambda_{PC} \mathcal{L}_{PC} + \lambda_{DC} \mathcal{L}_{DC} \\ & + \lambda_{SSIM} \mathcal{L}_{SSIM} + \lambda_{Smooth} \mathcal{L}_{Smooth}, \end{aligned} \quad (4)$$

where two commonly used regularization terms for depth estimation are also applied, including structural similarity  $\mathcal{L}_{SSIM}$  [25] and depth smoothness  $\mathcal{L}_{Smooth}$  [20, 14]. The weights are empirically set as:  $\lambda_{PC} = 0.8$ ,  $\lambda_{DC} = 0.1$ ,  $\lambda_{SSIM} = 0.2$ ,  $\lambda_{Smooth} = 0.0067$ .

### 3.2. Memory-efficient Design

One major limitation of deep learning based MVS methods is scalability: the memory-intensive 3D-CNN renders the learned MVS difficult to apply to high-resolution scenarios. This limitation is especially pronounced in the current self-supervised MVS framework [27, 3, 28], which almost doubles memory usage. As shown in Figure 1(a),

the network needs to perform forward inference and reverse gradient calculation for both the weak and strong augmentation branches. In this work, we redesign the E2E self-supervised MVS framework, which effectively reduces memory consumption and improves the model performance.

Formally, we freeze the weak-augmentation branch and apply photometric consistency and depth consistency to the output of the strong-augmentation branch, as shown in Figure 3. The weak-augmentation branch only conducts forward operations to prepare pseudo-depth maps, reducing GPU memory usage by 43%, from 14100 MiB to 8000 MiB as reported in Table 1. The results in Table 1 also reveal that applying photometric consistency to the strong-augmentation branch benefits model performance, improving the overall metric from 0.3483 to 0.3371 on the DTU dataset. The considerable performance gain mainly comes from supplying more diverse augmentation samples for photometric consistency. Another experiment also confirms the efficacy of providing data augmentation samples for photometric consistency, as shown in Table 2, where the interference of depth consistency is excluded.

### 3.3. Asymmetric View Selection Policy

Several source views  $I_{2 \rightarrow N}$  are selected to provide geometric information when predicting the depth map of the reference view  $I_1$ . Following MVSNet, recent self-supervised MVS methods select source views with top- $(N - 1)$  view-scores [33] for the weak and strong augmentation branches. We name this the top- $K$  view selection policy. The top- $K$  policy selects those source views with the optimal viewpoint difference from the reference view, leading to more accurate predicted depth maps. However, sharing the same source view selection policy on weak and strong branches limits the sample diversity between the two branches.

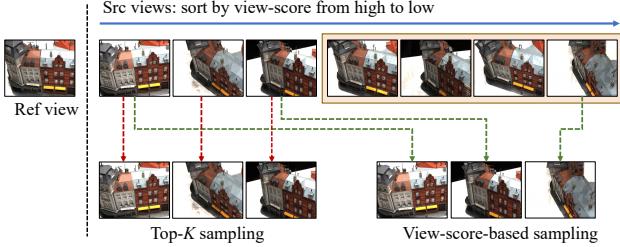


Figure 4: Depiction of view-score-based sampling. Compared to the top- $K$  sampling, our view-score-based sampling can select any source view sample, with a higher probability of selecting views with high view-scores.

In this work, we explore an asymmetric view selection policy to leverage weak-strong augmentation branches more efficiently and boost performance. Specifically, for the weak-augmentation branch, we maintain the top- $K$  view selection policy to provide accurate pseudo-depth maps. For the strong-augmentation branch, we propose a view-score-based selection policy. The view-score-based policy is designed based on two considerations. First, compared with the top- $K$  policy, our approach can select any source view. In this way, for a given reference view, the scope of the depth consistency constraint extends from color augmentation samples to viewpoint diversity samples. Second, compared with the random sampling policy, we assign higher selection probabilities to source views with higher view-scores, making training more stable and achieving higher performance. Experiments in Table 3 verify that the random sampling policy improves performance by introducing more diverse source view samples, while our view-score-based policy acquires the best performance. Figure 4 depicts our view-score-based sampling. To be specific, for the strong-augmentation branch, the  $N - 1$  source views are selected from all source views according to their view-scores (i.e., views with higher view-scores have a greater probability of being selected), where the view-score takes the definition as MVSNet [37, 33].

### 3.4. Region-aware Depth Consistency

Depth consistency plays an important role in end-to-end self-supervised MVS methods, where the output of the weak-augmentation branch is employed as the pseudo-depth map to supervise the learning of the strong-augmentation branch. However, as shown in Figure 2, the depth consistency loss suffers from imbalance among pixels, where texture-less backgrounds or object boundaries (with sudden changes in depth) usually lead to a significant increase in depth consistency loss. These regions usually correspond to erroneous pseudo-depths where predictions are inaccurate. U-MVSNet [28] selects to filter out those unreliable pseudo-depths through the uncertainty maps obtained by Monte-Carlo Dropout [13]. In contrast, as shown

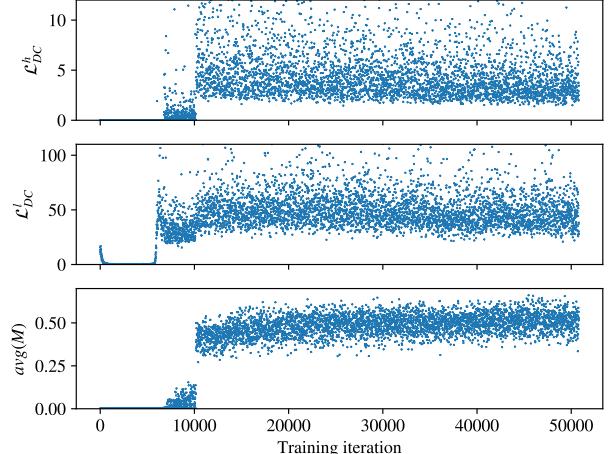


Figure 5: Partition visualization for depth consistency loss based on online cross-view checking.  $\mathcal{L}_{DC}^h$  denotes the depth consistency loss for high-quality regions.  $\mathcal{L}_{DC}^l$  indicates depth consistency in low-quality regions.  $avg(M)$  represents the ratio of high-quality pseudo-depths in the entire pseudo-depth map.

in Table 4, we experimentally find that even those erroneous pseudo-depths play a significant positive role in depth consistency. A more reasonable solution is to keep accurate and erroneous pseudo-depths, while each employs different loss weights to alleviate the imbalance in depth consistency. In this work, we propose alleviating the imbalance in depth consistency via region-aware loss weights.

Formally, as shown in Figure 3, we use a dynamic depth gallery to cache and update pseudo-depth maps for all reference views during the training phase. These cached pseudo-depth maps are then utilized to separate the pseudo-depth maps into high-quality and low-quality regions with online cross-view checking. Taking an arbitrary pixel  $\mathbf{p}_{1,k}$  in the reference view as an example, we first cast the 2D point to the corresponding 3D point  $\mathbf{P}_{1,k}$  with the pseudo-depth value  $D_1^w(\mathbf{p}_{1,k})$  (from the prediction of the weak-augmentation branch). Then, we project the 3D point to the  $i$ -th source view with the camera intrinsics and extrinsics, getting the 2D point  $\mathbf{p}_{i,k}$ . Finally, with the pseudo-depth of  $D_i^w(\mathbf{p}_{i,k})$  (from cached depth maps), we re-project the 2D point  $\mathbf{p}_{i,k}$  back to the reference view, and getting the projected 2D point  $\hat{\mathbf{p}}_{1,k}$  and depth value  $\hat{D}_1(\hat{\mathbf{p}}_{1,k})$  in the reference view. By defining the re-projection error in pixels and depth as  $e_{pixel} = \|\mathbf{p}_{1,k} - \hat{\mathbf{p}}_{1,k}\|_2$  and  $e_{depth} = \|\hat{D}_1(\hat{\mathbf{p}}_{1,k}) - D_1^w(\mathbf{p}_{1,k})\|_1 / \|D_1^w(\mathbf{p}_{1,k})\|_1$ , the high-quality mask  $M_1$  of the reference view is computed as

$$M_{1,k} = (C_{1,k}^w > \tau_1) \\ \left[ \left( \sum_{i=2}^S (e_{pixel} < \tau_2) \cdot (e_{depth} < \tau_3) \right) \geq \tau_4 \right], \quad (5)$$

where  $k$  is the index of the pixel,  $C_1^w$  is the predicted confidence map, and  $S$  is the total number of source views (10 by default). Hyperparameters  $\tau_1, \tau_2, \tau_3$ , and  $\tau_4$  are default set to 0.5, 0.5, 0.01, and 4.

With the high-quality mask computed in Equation 5, the pseudo-depth map  $D_1^w$  is partitioned into high-quality region  $D_1^{w,h} = D_1^w \odot M_1$  and low-quality region  $D_1^{w,l} = D_1^w \odot (1 - M_1)$ , and we reformulate the training loss of Equation 4 as

$$\begin{aligned} \mathcal{L} = & \lambda_{PC} \mathcal{L}_{PC} + \lambda_{DC}^h \mathcal{L}_{DC}^h + \lambda_{DC}^l \mathcal{L}_{DC}^l \\ & + \lambda_{SSIM} \mathcal{L}_{SSIM} + \lambda_{Smooth} \mathcal{L}_{Smooth}, \end{aligned} \quad (6)$$

where  $\lambda_{DC}^h$  and  $\lambda_{DC}^l$  denote the depth consistency loss weights for high-quality and low-quality regions, respectively. As shown in Figure 5, our proposed online cross-view checking can effectively partition depth consistency, where the consistency loss of partitioned low-quality regions  $\mathcal{L}_{DC}^l$  is an order of magnitude higher than  $\mathcal{L}_{DC}^h$  of high-quality regions. Reasonably setting the loss weights  $\lambda_{DC}^h$  and  $\lambda_{DC}^l$  of these two regions will maximize the impact of depth consistency, which is the motivation of our proposed region-aware depth consistency.

## 4. Experiments

### 4.1. Datasets

**DTU** [2] is an indoor dataset with multi-view images and corresponding camera parameters. There are 124 scenes scanned from 49 or 64 views under seven lighting conditions. We follow the setup of MVSNet [33] to divide the training, validation, and evaluation sets. In the DTU benchmark, the model is trained on the training set and tested on the evaluation set. We employ official error metrics from DTU to evaluate *Accuracy*, *Completeness*, and *Overall*.

**Tanks&Temples** [15] is a large-scale dataset containing various outdoor scenes. It includes an intermediate subset and an advanced subset. This benchmark is evaluated online by submitting the generated point cloud to the official website. In this benchmark, the F-score is calculated for each scene, and we separately report the average F-scores for the intermediate and advanced subsets.

### 4.2. Implementation Details

**Training Details.** The proposed ES-MVSNet is trained on the DTU [2] dataset. Following [6, 10, 14, 27, 3], we use the high-resolution DTU data provided by the open source code of MVSNet [33]. We first resize the input images to  $600 \times 800$ , following previous methods. Then we crop resized images into  $512 \times 640$  patches. The number of images  $N$  is set to 5 and 4 for the weak and strong augmentation branches, respectively. Data augmentation strategies are the same as JDACS [27], including gamma correction

| M. design | Acc( $\downarrow$ ) | Comp( $\downarrow$ ) | Overall( $\downarrow$ ) | GPU memory [MiB]      |
|-----------|---------------------|----------------------|-------------------------|-----------------------|
|           | 0.3762              | 0.3203               | 0.3483                  | 14100                 |
| ✓         | 0.3582              | 0.3160               | <b>0.3371</b>           | 8000 ( $\downarrow$ ) |

Table 1: Ablation study of memory-efficient design. “M. design” denotes memory-efficient design. The loss weight  $\lambda_{DC}$  is set to 0.1 by default.

| $\lambda_{DC}$ | Augmentation | Acc( $\downarrow$ ) | Comp( $\downarrow$ ) | Overall( $\downarrow$ ) |
|----------------|--------------|---------------------|----------------------|-------------------------|
| 0              | None         | 0.3842              | 0.3510               | 0.3676                  |
|                | Color        | 0.3594              | 0.3340               | <b>0.3467</b>           |

Table 2: The impact of data augmentation on photometric consistency. The loss weight  $\lambda_{DC}$  is set to 0 to exclude the affect of depth consistency.

and color jitter. We adopt the backbone of Cas-MVSNet [8] to construct our multi-scale pipeline with 3 stages. For each stage, we use different feature maps and the 3D-CNN network parameters. The whole network is optimized by an Adam optimizer in Pytorch for 15 epochs with an initial learning rate of 0.0001, which is downscaled by a factor of 2 after 10, 12, and 14 epochs. We train with a batch size of 8 using eight NVIDIA V100 GPUs.

**Testing Details.** The model trained on DTU training set is used for testing on DTU testing set. The input image number  $N$  is set to 5, each with a resolution of  $1152 \times 1600$ . The model trained on DTU training dataset is directly used for testing on Tanks&Temples intermediate and advanced datasets without finetuning. The image sizes are set to  $1024 \times 1920$  or  $1024 \times 2048$  and the input image number  $N$  is set to 7. When fusing the per-view depth maps into the final point cloud, we employ the photometric and geometric consistencies, and the hyper-parameter settings are consistent with RC-MVSNet [3].

### 4.3. Ablation Study

**Effect of memory-efficient design.** Our memory-efficient design freezes the parameters of the weak-augmentation branch and applies photometric consistency to the output of the strong-augmentation branch. As shown in Table 1, this small design reduces the memory usage of the model by 43%, which is conducive to the application of the model in high-resolution scenarios. Table 1 also reveals that the model performance is significantly improved, thanks to the photometric consistency applied on more diverse samples via data augmentation. The additional experiments in Table 2, where the depth consistency is excluded by setting loss weight  $\lambda_{DC}$  to 0, also confirm that applying photometric consistency to augmented samples helps improve the performance of self-supervised MVS. In the following experiments, we employ the proposed memory-efficient framework by default.



Figure 6: Qualitative results on some scenes of DTU dataset [2].



Figure 7: Qualitative results on scenes of Tanks&Temples dataset [15]. The model is trained on the DTU training set only.

| W. policy        | S. policy        | Acc( $\downarrow$ ) | Comp( $\downarrow$ ) | Overall( $\downarrow$ ) | Memory [MiB] |
|------------------|------------------|---------------------|----------------------|-------------------------|--------------|
| Top- $K$         | Top- $K$         | 0.3582              | 0.3160               | 0.3371                  | 8000         |
|                  | Random           | 0.3525              | 0.3086               | 0.3306                  | 8000         |
|                  | View-score-based | 0.3520              | 0.3004               | <b>0.3262</b>           | 8000         |
| View-score-based | View-score-based | 0.3545              | 0.2996               | 0.3271                  | 8000         |

Table 3: Ablation study of source view selection policy. “W. policy”: the view selection policy for the weak-augmentation branch. “S. policy”: the view selection policy for the strong-augmentation branch. The loss weight  $\lambda_{DC}$  is set to 0.1 by default.

**Effect of asymmetric view selection policy.** In this section, we investigate the impact of the asymmetric view selection policy. We first fix the view selection policy of the weak-augmentation branch as “top- $K$ ” to provide high-quality pseudo-depth maps. Then we vary the policy of the strong-augmentation branch. The experimental results in Table 3 show that the model performance improves when switching the strong-augmentation branch’s policy from “top- $K$ ” to “random sampling”. Furthermore, the model performs best when adopting a sampling policy based on view-scores. Compared with random sampling, the view-score-based sampling can effectively reduce the influence of those source views with significant viewpoint differences from the reference view.

**Effect of region-aware depth consistency.** As verified in Figure 5, the depth consistency loss of high-quality and low-quality regions differs by nearly an order of magnitude, making it unreasonable to utilize a single loss weight  $\lambda_{DC}$ . In this section, we experimentally verify the effectiveness of

| $\lambda_{DC}$ | $\lambda_{DC}^h$ | $\lambda_{DC}^l$ | Acc( $\downarrow$ ) | Comp( $\downarrow$ ) | Overall( $\downarrow$ ) | GPU [MiB] |
|----------------|------------------|------------------|---------------------|----------------------|-------------------------|-----------|
| 0.1            | 0.               | 0.               | 0.3520              | 0.3004               | 0.3262                  | 8000      |
|                |                  | 0.1              | 0.3605              | 0.3234               | 0.3420                  | 8000      |
|                |                  | 0.5              | 0.3459              | 0.3086               | 0.3291                  | 8000      |
|                |                  | 0.5              | 0.3550              | 0.2961               | 0.3255                  | 8000      |
| 0.5            | 0.1              | 0.3479           | 0.2983              | <b>0.3231</b>        | 8000                    |           |

Table 4: Ablation study of region-aware depth consistency. For the view selection policy, the strong-branch utilizes view-score-based sampling by default.

| Overall( $\downarrow$ ) \ $\lambda_{DC}^h$ | 0      | 0.1    | 0.5           | 1.0    |
|--|--------|--------|---------------|--------|
| $\lambda_{DC}^l$                           | 0      | 0.3420 | 0.3280        | 0.3255 |
| 0  | 0.3420 | 0.3280 | 0.3255        | 0.3336 |
| 0.1  | 0.3291 | 0.3262 | <b>0.3231</b> | 0.3266 |
| 0.2  | 0.3329 | 0.3308 | 0.3275        | 0.3344 |
| 0.4  | 0.3481 | 0.3363 | 0.3357        | 0.3370 |

Table 5: Detailed ablation of loss weights  $\lambda_{DC}^h$  and  $\lambda_{DC}^l$ .

the proposed region-aware depth consistency. The pseudo-depth maps are partitioned into high-quality and low-quality regions through online cross-view checking. Then these two regions adopt loss weights  $\lambda_{DC}^h$  and  $\lambda_{DC}^l$ , respectively. As shown in Table 4, compared with no depth consistency (Overall metric of 0.3420), applying depth consistency independently to high-quality or low-quality regions has a significant positive effect. For example, when setting low-quality regions’ depth consistency loss weight  $\lambda_{DC}^l$  to 0.1, the Overall metric improves from 0.3420 to 0.3291. We believe that the key to performance improvement lies in the balance of the accurate and the erroneous pseudo-depth consistencies. When the loss weight of  $\lambda_{DC}^h$  is set to 0.5,

| M. design | A. policy | R.D. | Acc(↓) | Comp(↓) | Overall(↓)    | GPU [MiB] |
|-----------|-----------|------|--------|---------|---------------|-----------|
|           |           |      | 0.3762 | 0.3203  | 0.3483        | 14100     |
| ✓         |           |      | 0.3582 | 0.3160  | 0.3371        | 8000      |
| ✓         | ✓         |      | 0.3520 | 0.3004  | 0.3262        | 8000      |
| ✓         | ✓         | ✓    | 0.3479 | 0.2983  | <b>0.3231</b> | 8000      |

Table 6: Ablation study of different components of our proposed self-supervised MVS framework. “M. design”: memory-efficient design. “A. policy”: asymmetric view selection policy. “R.D.”: region-aware depth consistency.

|                       | Method                    | Acc(↓) | Comp(↓) | Overall(↓)   |
|-----------------------|---------------------------|--------|---------|--------------|
| Supervised            | SurfaceNet [11]           | 0.450  | 1.040   | 0.745        |
|                       | MVSNet [33]               | 0.396  | 0.527   | 0.462        |
|                       | Cas-MVSNet [8]            | 0.325  | 0.385   | 0.355        |
|                       | PatchmatchNet [24]        | 0.427  | 0.277   | 0.352        |
|                       | CVP-MVSNet [32]           | 0.296  | 0.406   | 0.351        |
|                       | UCSNet [4]                | 0.338  | 0.349   | 0.344        |
|                       | UniMVSNet [19]            | 0.352  | 0.278   | 0.315        |
|                       | GBi-Net [17]              | 0.315  | 0.262   | 0.289        |
| Multi-Stage Self-Sup. | Self\_sup CVP-MVSNet [31] | 0.308  | 0.418   | 0.363        |
|                       | U-MVSNet [28]             | 0.354  | 0.3535  | 0.354        |
|                       | KD-MVS [7]                | 0.359  | 0.295   | 0.327        |
| E2E Self-Sup.         | Unsup_MVSNet [14]         | 0.881  | 1.073   | 0.977        |
|                       | MVS2 [6]                  | 0.760  | 0.515   | 0.637        |
|                       | M3VSN [10]                | 0.636  | 0.531   | 0.583        |
|                       | DS-MVSNet [16]            | 0.374  | 0.347   | 0.361        |
|                       | JDACS-MS [27]             | 0.398  | 0.318   | 0.358        |
|                       | RC-MVSNet [3]             | 0.396  | 0.295   | 0.345        |
|                       | <b>ES-MVSNet (ours)</b>   | 0.348  | 0.298   | <b>0.323</b> |

Table 7: Point cloud evaluation results on DTU dataset [2]. The lower is better. The sections are partitioned into supervised, multi-stage self-supervised, and end-to-end self-supervised, respectively. All the results other than ours are from previously published literature.

and the loss weight of  $\lambda_{DC}^l$  is set to 0.1, the model achieves the best performance of 0.3231. The detailed ablation of loss weights  $\lambda_{DC}^h$  and  $\lambda_{DC}^l$  is shown in Table 5.

**Effect of each component.** To evaluate the performance gain of each component in our approach, we provide factor-by-factor ablation experiments, as shown in Table 6. We can clearly observe that the three key designs of our approach all yield better reconstruction results, and our framework significantly reduces GPU memory usage.

#### 4.4. Benchmark Performance

**Evaluation on DTU Dataset.** We evaluate the depth prediction performance on the DTU test set, and compare with previous state-of-the-art methods in Table 7. Our ES-MVSNet architecture achieves the best overall score (lower is better) among all end-to-end self-supervised MVS methods. Our model improves the overall score from 0.345 of RC-MVSNet [3] to 0.323. The overall score is also better than the multi-stage self-supervised approaches, and even most supervised methods. Visualizations of our point cloud reconstruction results can be found in Figure 6.

|                       | Method                    | Mean ↑       |
|-----------------------|---------------------------|--------------|
| Supervised            | MVSNet [33]               | 43.48        |
|                       | CIDER [29]                | 46.76        |
|                       | PatchmatchNet [24]        | 53.15        |
|                       | CVP-MVSNet [32]           | 54.03        |
|                       | UCSNet [4]                | 54.83        |
|                       | Cas-MVSNet [8]            | 56.42        |
|                       | AA-RMVSNet [26]           | 61.51        |
|                       | UniMVSNet [19]            | 64.36        |
| Multi-Stage Self-Sup. | Self\_sup CVP-MVSNet [31] | 46.71        |
|                       | U-MVSNet [28]             | 57.15        |
|                       | KD-MVS [7]                | ‡64.14       |
| E2E Self-Sup.         | MVS2 [6]                  | 37.21        |
|                       | M3VSN [10]                | 37.67        |
|                       | JDACS-MS [27]             | 45.48        |
|                       | DS-MVSNet [16]            | 54.76        |
|                       | RC-MVSNet [3]             | 55.04        |
|                       | <b>ES-MVSNet (ours)</b>   | <b>58.10</b> |

Table 8: Point cloud evaluation results on the *intermediate* subsets of Tanks&Temples dataset [15]. Higher scores are better. The Mean is the average score of all scenes. ‡ denotes further finetuning on the BlendedMVS [35] dataset.

|                       | Method                  | Mean↑        |
|-----------------------|-------------------------|--------------|
| Supervised            | CIDER [29]              | 23.12        |
|                       | R-MVSNet [34]           | 24.91        |
|                       | CasMVSNet [8]           | 31.12        |
|                       | PatchmatchNet [24]      | 32.31        |
|                       | AA-RMVSNet [26]         | 33.53        |
|                       | UniMVSNet [19]          | 38.96        |
| Multi-Stage Self-Sup. | U-MVSNet [28]           | 30.97        |
|                       | KD-MVS [7]              | ‡37.96       |
| E2E Self-Sup.         | RC-MVSNet [3]           | 30.82        |
|                       | <b>ES-MVSNet (ours)</b> | <b>34.90</b> |

Table 9: Point cloud evaluation results on the *advanced* subset of Tanks&Temples dataset [15]. Higher scores are better. The Mean is the average score of all scenes. ‡ denotes further finetuning on the BlendedMVS [35] dataset.

**Evaluation on Tanks&Temples.** We train the proposed ES-MVSNet on the DTU training set and test it on the Tanks&Temples dataset without fine-tuning. We compare our method with state-of-the-art supervised, pseudo-label-based multi-stage self-supervised and end-to-end self-supervised approaches. Tables 8 and 9 show the evaluation results for the intermediate and advanced subsets, respectively. Our ES-MVSNet achieves the best performance among end-to-end self-supervised methods on both subsets. The anonymous evaluation on the leaderboard [1] is named ES-MVSNet. The qualitative point cloud reconstruction results are visualized in Figure 7. More detailed comparisons of each scene are reported in our supplementary materials.

## 5. Conclusions

In this work, we propose an efficient E2E self-supervised MVS framework, named ES-MVSNet, which achieves state-of-the-art performance without the need for additional third-party models. The ES-MVSNet model includes three

key designs: a memory-efficient structure, an asymmetric view selection policy, and a region-aware depth consistency.

## References

- [1] Tanks and temples leaderboard. <https://www.tanksandtemples.org/leaderboard>.
- [2] Henrik Aanaes, Rasmus Ramsboel Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *Int. J. Comput. Vis.*, 120(2):153–168, 2016.
- [3] Di Chang, Aljaz Bozic, Tong Zhang, Qingsong Yan, Yingcong Chen, Sabine Süsstrunk, and Matthias Nießner. Rc-mvsnet: Unsupervised multi-view stereo with neural rendering. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *ECCV*, pages 665–680, 2022.
- [4] Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. Deep stereo using adaptive thin volume representation with uncertainty awareness. In *CVPR*, pages 2521–2531, 2020.
- [5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [6] Yuchao Dai, Zhidong Zhu, Zhibo Rao, and Bo Li. MVS2: deep unsupervised multi-view stereo with multi-view symmetry. In *2019 International Conference on 3D Vision, 3DV 2019, Québec City, QC, Canada, September 16-19, 2019*, pages 1–8. IEEE, 2019.
- [7] Yikang Ding, Qingtian Zhu, Xiangyue Liu, Wentao Yuan, Haotian Zhang, and Chi Zhang. KD-MVS: knowledge distillation based self-supervised learning for multi-view stereo. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *ECCV*, volume 13691 of *Lecture Notes in Computer Science*, pages 630–646. Springer, 2022.
- [8] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *CVPR*, pages 2492–2501, 2020.
- [9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.
- [10] Baichuan Huang, Hongwei Yi, Can Huang, Yijia He, Jingbin Liu, and Xiao Liu. M3VSNET: unsupervised multi-metric multi-view stereo network. In *2021 IEEE International Conference on Image Processing, ICIP 2021, Anchorage, AK, USA, September 19-22, 2021*, pages 3163–3167. IEEE, 2021.
- [11] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *ICCV*, pages 2326–2334. IEEE Computer Society, 2017.
- [12] Mengqi Ji, Jinzhi Zhang, Qionghai Dai, and Lu Fang. Surfacenet+: An end-to-end 3d neural network for very sparse multi-view stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(11):4078–4093, 2021.
- [13] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017.
- [14] Tejas Khot, Shubham Agrawal, Shubham Tulsiani, Christoph Mertz, Simon Lucey, and Martial Hebert. Learning unsupervised multi-view stereopsis via robust photometric consistency. *CoRR*, abs/1905.02706, 2019.
- [15] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: benchmarking large-scale scene reconstruction. *ACM Trans. Graph.*, 36(4):78:1–78:13, 2017.
- [16] Jingliang Li, Zhengda Lu, Yiqun Wang, Ying Wang, and Jun Xiao. Ds-mvsnet: Unsupervised multi-view stereo via depth synthesis. *MM ’22*, New York, NY, USA, 2022. Association for Computing Machinery.
- [17] Zhenxing Mi, Di Chang, and Dan Xu. Generalized binary search network for highly-efficient multi-view stereo. In *CVPR*, pages 12981–12990. IEEE, 2022.
- [18] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, volume 12346 of *Lecture Notes in Computer Science*, pages 405–421. Springer, 2020.
- [19] Rui Peng, Rongjie Wang, Zhenyu Wang, Yawen Lai, and Ronggang Wang. Rethinking depth estimation for multi-view stereo: A unified representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 8635–8644. IEEE, 2022.
- [20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *ICML*, volume 139, pages 8748–8763, 2021.
- [21] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113. IEEE Computer Society, 2016.
- [22] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *ECCV*, volume 9907 of *Lecture Notes in Computer Science*, pages 501–518. Springer, 2016.
- [23] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, pages 519–528. IEEE Computer Society, 2006.
- [24] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned

- multi-view patchmatch stereo. In *CVPR*, pages 14194–14203, 2021.
- [25] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.
- [26] Zizhuang Wei, Qingtian Zhu, Chen Min, Yisong Chen, and Guoping Wang. Aa-rmvsnet: Adaptive aggregation recurrent multi-view stereo network. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 6167–6176. IEEE, 2021.
- [27] Hongbin Xu, Zhipeng Zhou, Yu Qiao, Wenxiong Kang, and Qiuxia Wu. Self-supervised multi-view stereo via effective co-segmentation and data-augmentation. In *AAAI*, pages 3030–3038, 2021.
- [28] Hongbin Xu, Zhipeng Zhou, Yali Wang, Wenxiong Kang, Baigui Sun, Hao Li, and Yu Qiao. Digging into uncertainty in self-supervised multi-view stereo. In *ICCV*, pages 6058–6067. IEEE, 2021.
- [29] Qingshan Xu and Wenbing Tao. Learning inverse depth regression for multi-view stereo with correlation cost volume. In *AAAI*, pages 12508–12515. AAAI Press, 2020.
- [30] Jianfeng Yan, Zizhuang Wei, Hongwei Yi, Mingyu Ding, Runze Zhang, Yisong Chen, Guoping Wang, and Yu-Wing Tai. Dense hybrid recurrent multi-view stereo net with dynamic consistency checking. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, volume 12349 of *Lecture Notes in Computer Science*, pages 674–689. Springer, 2020.
- [31] Jiayu Yang, Jose M. Alvarez, and Miaomiao Liu. Self-supervised learning of depth inference for multi-view stereo. In *CVPR*, pages 7526–7534. Computer Vision Foundation / IEEE, 2021.
- [32] Jiayu Yang, Wei Mao, Jose M. Alvarez, and Miaomiao Liu. Cost volume pyramid based depth inference for multi-view stereo. In *CVPR*, pages 4876–4885. Computer Vision Foundation / IEEE, 2020.
- [33] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV*, volume 11212, pages 785–801. Springer, 2018.
- [34] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *CVPR*, pages 5525–5534. Computer Vision Foundation / IEEE, 2019.
- [35] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *CVPR*, pages 1787–1796. Computer Vision Foundation / IEEE, 2020.
- [36] Hongwei Yi, Zizhuang Wei, Mingyu Ding, Runze Zhang, Yisong Chen, Guoping Wang, and Yu-Wing Tai. Pyramid multi-view stereo net with self-adaptive view aggregation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, pages 766–782, 2020.
- [37] Runze Zhang, Shiwei Li, Tian Fang, Siyu Zhu, and Long Quan. Joint camera clustering and surface segmentation for large-scale multi-view stereo. In *ICCV*, pages 2084–2092. IEEE Computer Society, 2015.