MVSFormer: Multi-View Stereo by Learning Robust Image Features and Temperature-based Depth

Anonymous authors
Paper under double-blind review

Abstract

Feature representation learning is the key recipe for learning-based Multi-View Stereo (MVS). As the common feature extractor of learning-based MVS, vanilla Feature Pyramid Networks (FPNs) suffer from discouraged feature representations for reflection and texture-less areas, which limits the generalization of MVS. Even FPNs worked with pre-trained Convolutional Neural Networks (CNNs) fail to tackle these issues. On the other hand, Vision Transformers (ViTs) have achieved prominent success in many 2D vision tasks. Thus we ask whether ViTs can facilitate feature learning in MVS? In this paper, we propose a pre-trained ViT enhanced MVS network called MVSFormer, which can learn more reliable feature representations benefited by informative priors from ViT. Then MVSFormer-P and MVSFormer-H are further proposed with frozen ViT weights and trainable ones respectively. MVSFormer-P is more efficient while MVSFormer-H can achieve superior performance. MVSFormer can be generalized to various input resolutions with efficient multi-scale training strengthened by gradient accumulation. Moreover, we discuss the merits and drawbacks of classification and regression-based MVS methods, and further propose to unify them with a temperature-based strategy. MVSFormer achieves state-of-the-art performance on the DTU dataset. Particularly, our anonymous Tanks-and-Temples submission of MVSFormer was submitted in May/2022. And until the day of our paper is submitted to TMLR, our results are still ranked as Top-1 on both intermediate and advanced sets of the highly competitive Tanks-and-Temples leaderboard. Codes and models will be released upon acceptance.

1 Instruction

Multi-View Stereo (MVS) aims to reconstruct highly detailed 3D representations with multi-view images, with the key step of estimating depth maps with known camera poses Furukawa & Hernández (2015). Many traditional methods Barnes et al. (2009); Furukawa & Ponce (2009); Galliani et al. (2015); Schönberger et al. (2016) successfully make use of matching low-level features of images; unfortunately they may be negatively affected by various occlusions and different illumination conditions. To this end, learning-based methods enhanced by Deep Neural Networks (DNNs) Ji et al. (2017); Yao et al. (2018); Gu et al. (2020); Giang et al. (2021); Wei et al. (2021) have developed recently. Typically, there are three steps for learning-based MVS methods, i.e., feature extraction, cost volume construction, and cost volume regularization Yao et al. (2018); Giang et al. (2021).

Many works devote to formulating better cost volumes with efficient multi-stage models Gu et al. (2020); Yang et al. (2020); Mi et al. (2021) and visibility information Zhang et al. (2020); Xu & Tao (2020). Meanwhile, learning a superior cost volume regularization with hybrid 3D U-Net structures Luo et al. (2019); Sormann et al. (2020) or Recurrent Neural Networks (RNNs) Yao et al. (2019); Wei et al. (2021, 2022) is also shown to improve performance. Generally, the cost volume regularization is designed to refine noise-contaminated cost volumes with non-Lambertian surfaces or object occlusions Yao et al. (2018) to smooth feature correlations as shown in Fig. 1(c)(e). Such regularization can not completely rectify ambiguous feature matchings from reflections or texture-less regions with unreliable 2D image features. Therefore, it is still of great significance to learn good representative features during the feature extraction to improve the generalization of MVS.
As a common solution for feature extraction, Feature Pyramid Network (FPN) [Lin et al. (2017)] learns multi-scale image features in most MVS networks. Some works leverage deformable convolutions [Wang et al. (2021b); Wei et al. (2021); Ding et al. (2021)], attention mechanisms [Yi et al. (2020); Zhu et al. (2021); Ding et al. (2021)], and normal curvatures [Xu et al. (2020); Giang et al. (2021)] to learn more reliable features for FPNs. Nevertheless, these works still suffer from poor generalization in modeling the reflection and texture-less regions as visualized in Fig. 1(c)(d). On the other hand, few previous efforts have explicitly exploring the features from the Convolutional Neural Networks (CNNs) pre-trained on extra image data [Ding et al. (2022), e.g., ResNet] as such pre-trained CNNs may have some problems in MVS: 1) low-level features of CNNs only consider limited receptive fields, which lack the holistic image understanding, and fail to tackle with reflections and texture-less areas; 2) high-level features of CNNs are of highly semantic abstract, thus best for the classification rather than the fine-grained visual feature matching. We empirically validate that pre-trained CNN models fail to achieve significant improvement in MVS as in Tab. 3.

Recent Vision Transformers (ViTs) have achieved impressive performance on various image understanding tasks [Dosovitskiy et al. (2020); Caron et al. (2021); Wang et al. (2021c); Liu et al. (2021); Chu et al. (2021a); He et al. (2021)]. Thus, a nature question is whether we can significantly strengthen the feature representation learning of MVS with pre-trained transformers from external 2D image dataset? For the issues of reflections and texture-less regions in MVS, ViTs equipped with long-range attention modules can provide global understanding for MVS models rather than the low-level textures. Moreover, the patch-wise feature encoding of ViTs works reasonably well for feature matching [Sun et al. (2021); Jiang et al. (2021)]. Since the depth prediction is intrinsically a 1D feature matching problem along epipolar lines [Furukawa & Hernández (2015)], ViTs shall be the recipe for learning-based MVS. Unfortunately, to the best of our knowledge, there is no work explicitly exploiting pre-trained ViTs for MVS.

To this end, we thoroughly explore how to make pre-trained ViTs boost the MVS performance. As shown in Fig. 1(e)(f), features from pre-trained ViT are complementary to those from FPN, and facilitate better modeling reflection and texture-less regions in MVS. Formally, we propose using the pre-trained ViTs to enhance FPN for feature extraction in MVS, and further formulate a novel MVS Transformer (MVSFormer). Specifically, by considering two typical ViT backbones – plain-ViT [Dosovitskiy et al. (2020); Caron et al. (2021); He et al. (2021)] and hierarchical-ViT [Wang et al. (2021c); Liu et al. (2021); Chu et al. (2021a)], we implement MVSFormer-P and MVSFormer-H as in Fig. 2(A). To reduce the computation cost for MVSFormer-P, we take the frozen off-the-shelf ViT backbone pre-trained by the no label self-supervised method – DINO [Caron et al. (2021)] with the good clustering property. We employ the hierarchical ViT Twins [Chu et al. (2021a)] as the backbone of MVSFormer-H. Benefited by the pyramid architecture and the efficient attention mechanism, MVSFormer-H can be trained in high-resolution and typically achieve better results compared with MVSFormer-P.

---

1Intuitively Twins can also be pre-trained by the self-supervised task as DINO. However, it is non-trivial to implement and train it, due to technical difficulty and expensive running costs on ImageNet (taking 1 months on 2 V100 GPUs). Thus the self-supervised Twins is beyond the scope of this paper.
Further, we present an efficient multi-scale strategy to train MVSFormer, as it is non-trivial to directly train ViTs for MVS. In MVS tasks, the models should be tested on various high-resolution images Aanæs et al. (2016); Knapitsch et al. (2017), while they have to be trained on low-resolution to save the training computations. It is thus not amenable to directly extending pre-trained ViTs without scale invariance to MVS tasks. Our empirical experiments in Sec. 4.1 show that the proposed multi-scale strategy is sufficient to generalize MVSFormer to test images with much higher resolutions (1536 or 1920) compared with the largest one during the training (1280).

Furthermore, we technically unify the advantages of both regressive depth (regression), and argmax depth (classification) for MVSFormer. Different from previous works Wang et al. (2021a); Peng et al. (2022); Wang et al. (2022), we find that optimizing the MVS network with cross-entropy loss can achieve much more reliable confidence maps but slightly worse depth predictions. Because the argmax operation can not provide exact depth results, which harms the depth performance. To address this issue, MVSFormer predicts the temperature-based depth expectation instead of the argmax during the inference and achieves smooth depth predictions and superior final point clouds.

Our contributions can be highlighted as: (1) To the best of our knowledge, it is the first work that systematically explores the influence of pre-trained ViTs on MVS. Learning a better feature representation by the feature extractor is important to set up a bridge between 2D vision and 3D MVS tasks. (2) We propose a novel ViT enhanced MVS network – MVSFormer that has the implementations of MVSFormer-P and MVSFormer-H, which are further trained with the efficient multi-scale training strategy to be generalized for various resolutions. The proposed ViT pre-trained MVSFormer-H remarkably reduces the overall error of point cloud reconstruction in DTU from 0.312 to 0.289 compared with the CNN-based pre-trained ResNet with competitive computations and all other model settings unchanged as in Tab. 3. (3) We analyze the merits and limitations of regression and classification-based MVS, and propose a simple but effective way to unify both. Empirical qualitative evidences in Fig. 4 and Fig. 6 show that classification-based confidence can filter outliers for the real-world reconstruction. And quantitative results in Tab. 5 indicate that our temperature-based depth predictions also enjoy superior point cloud metrics. (4) The proposed methods can achieve the state-of-the-art performance in both DTU dataset Aanæs et al. (2016) and Tanks-and-Temples Knapitsch et al. (2017).

2 Related Works

Learning-based MVS methods. Learning-based MVS methods enhanced with DNNs have been extensively studied for MVS tasks Ji et al. (2017). MVSNet Yao et al. (2018) proposes an end-to-end pipeline based on a 2D CNN image feature extraction, cost volumes formulated by homography warping, and a cost volume regularization by 3D CNN. Many works are devoted to reducing the heavy computation of 3D CNN-based cost volume regularization with coarse-to-fine strategies Gu et al. (2020); Yang et al. (2020); Mi et al. (2021) and RNNs Yao et al. (2019); Wei et al. (2021); Mi et al. (2021); Wei et al. (2022). Meanwhile, some researches formulate a more reliable cost volume, such as the visibility of ViS-MVSNNet Zhang et al. (2020), and the epipolar aggregation of MVSTER Wang et al. (2022). Besides, many works try to learn a better cost regularization by hybrid 3D U-Net Luo et al. (2019), Sormann et al. (2020), RNN-3DCNN Wei et al. (2021), and epipolar attention Ma et al. (2021), Yang et al. (2021). Although many efforts are paid to achieve better cost volumes, we advocate that learning superior feature representations are more effective for a generalized MVS method. Our method explores using pre-trained ViTs to improve MVS feature learning, which is orthogonal to these works based on cost volumes.

Feature learning in MVS. Since 2D image features are critical in MVS learning, powerful FPN Lin et al. (2017a) is the common solution. FPN is designed as a U-Net to fuse multi-scale features. Deformation convolutions Dai et al. (2017) are widely used in MVS to improve the receptive fields flexibly Wang et al. (2021b); Wei et al. (2021); Mi et al. (2021); Ding et al. (2021). Furthermore, Xu et al. Xu et al. (2020) and Giang et al. Giang et al. (2021) leverage fixed and learnable normal curvatures to dynamically select kernel sizes for FPN, which can learn robust features for various image resolutions. Besides, the attention mechanism is also utilized for MVS feature learning. Many works leverage intra- or inter-attention during the feature learning Yi et al. (2020); Zhu et al. (2021); Ding et al. (2021) for long-range feature dependencies.
As these efforts exactly improve the performance of the FPN, we should notice that such features still have inductive biases from CNNs, thus leading to inferior performance to patch-wise ViTs [Dosovitskiy et al. (2020)]. There is no systematical exploration of pre-trained ViT models to MVS feature learning.

**Vision Transformers.** Self-supervised pre-trained transformers have achieved prominent success in Natural Language Processing (NLP) [Vaswani et al. (2017); Devlin et al. (2018); Brown et al. (2020)]. Inspired by these achievements, transformers are also introduced into the computer vision [Dosovitskiy et al. (2020); Liu et al. (2021); He et al. (2021)]. ViTs achieve state-of-the-art performance in many vision tasks, which include image classification [Dosovitskiy et al. (2020); Liu et al. (2021)], detection and segmentation [Zheng et al. (2021); Li et al. (2022)], and so on. Moreover, task specific transformers also achieve successes in optical flow [Huang et al. (2022)] and point cloud processing [Guo et al. (2021); Zhao et al. (2021)]. Compared with CNNs, ViTs enjoy much more long-range modelling capacity. Since ViTs lack some inductive biases inherent to CNNs, such as translation equivalence and locality, they are much more data-hungry to generalize to unseen data [Ascoli et al. (2021); Dosovitskiy et al. (2020); Xu et al. (2021)]. Therefore, pre-training is necessary for ViTs. For different pre-training tasks, ViTs can be categorized into self-supervised ones [Bao et al. (2021); He et al. (2021); Caron et al. (2021)] and supervised ones [Dosovitskiy et al. (2020); Liu et al. (2021); Chu et al. (2021a)]. Supervised ViTs are usually pre-trained for the classification task, while self-supervised ones are implemented with masked prediction [He et al. (2021); Bao et al. (2021)] or contrastive learning [Caron et al. (2021); Chen et al. (2021b)]. Furthermore, to reduce notorious computations and memory costs of the vanilla attention, many ViTs use multi-scale architectures [Liu et al. (2021); Wang et al. (2021c); Chu et al. (2021a)] instead of single-scale backbones. Particularly, many works tried to fine-tune pre-trained ViTs with task specific components, such as FPNs or CNNs, for various downstream tasks [Li et al. (2022); Huang et al. (2022); Chen et al. (2021a)]. Although pre-trained ViTs are appealing in many fields, releasing their potential in MVS is non-trivial, which will be discussed in the following sections.

**Multi-scale training.** It is well known that CNNs can be generalized to higher resolutions for high-level vision tasks such as classification [Chu et al. (2021b)], which is benefited from their scale invariance. Some classical object detection methods [Cai et al. (2016); Redmon & Farhadi (2017)] emphasize the importance to use multi-scale training for better robustness to various image scales. Further, some recent generation works [Dong et al. (2022); Chai et al. (2022); Cao et al. (2022)] show that multi-scale training is also a data and computation efficient way to improve the generation quality at high resolution. For the MVS learning, dynamic kernels enhanced by the curvature clues [Giang et al. (2021)] are leveraged to tackle the image scale gaps for CNN-based models. Unfortunately, the attention-based ViTs generally do not have the translation equivalence and locality; and these make ViTs more easily overfitting to specific resolution or sequence length than CNNs [Ascoli et al. (2021); Dosovitskiy et al. (2020); Xu et al. (2021)]. Although Conditional Positional Encoding (CPE) and zero-padding are utilized to alleviate the spatial overfitting of ViT [Chu et al. (2021b)], multi-scale training is still adopted in many ViTs for high-resolution detection tasks [Liu et al. (2021); Chu et al. (2021a); Carion et al. (2020)]. In this paper, we empirically evaluated that multi-scale training is the key strategy to ignite the ViTs for MVS in Sec. 4.2. Technically, we repurpose the gradient accumulation to improve GPU utilization for better efficiency of the multi-scale training.

### 3 Method

**Overview.** The architecture of MVSFormer is overviewed in Fig. 2, as the seminal MVS pipeline in Yao et al. (2018). We first briefly review pre-trained ViTs used in this paper. Given a group of \(N\) images with different views which contain reference image \(I_0 \in \mathbb{R}^{H \times W \times 3}\) and source images \(\{I_i\}_{i=1}^{N-1} \in \mathbb{R}^{H \times W \times 3}\), as well as the their camera intrinsics and extrinsics, MVSFormer learns feature representation in feature extraction (Sec. 3.1), enhanced by the plain ViT–DINO [Caron et al. (2021)] (Fig. 2(a)) or the hierarchical ViT–Twins [Chu et al. (2021a)] (Fig. 2(b)) with several novel training strategies (Sec. 3.2). Then the multi-stage cost volume formulation and regularization are presented to compute the probabilities of coarse-to-fine depth hypotheses (Sec. 3.3). Finally, cross-entropy loss is employed to optimize the MVSFormer, while the inference is made on depth expectation (Sec. 3.4).

**Preliminaries.** (1) **DINO** [Caron et al. (2021)] is pre-trained in a self-supervised way by the self-distillation of plain-ViT as in Fig. 2(a). The prominent characteristic of DINO is that attention maps of its last layer can
Figure 2: (A) Feature extractors of MVSFormer-P (a) and MVSFormer-H (b). Inputs for ViTs are downsampled to the 1/2 resolution. (B) Multi-scale cost volume formulation and regularization. ‘Warping’: warping source features with upsampled depth hypotheses (Eq. 3) for cost volumes (Eq. 4). ‘Volume Fusion’: fusing cost volumes from all source views with respective visibility (Eq. 5).

learn class-specific features leading to unsupervised object segmentations as discussed in Caron et al. (2021), and shown in Fig. 3. Thanks to the unsupervised training and the multi-crop strategy, the feature representations of DINO can be well generalized to various environments, illuminations, and resolutions. (2) Twins Chu et al. (2021a) is pre-trained in a supervised way by the hierarchical-ViT model as in Fig. 2(b). To further reduce the complexity, Twins employs the separable locally-grouped self-attention and global sub-sampled attention to construct each attention block. Such a global and local design outperforms the classic pyramid ViT Wang et al. (2021c). Twins also leverages CPE Chu et al. (2021b) with 2D depthwise convolutions instead of the absolute positional encoding in other ViTs Dosovitskiy et al. (2020).

3.1 Feature Extraction

As many MVS works, we also use FPN Lin et al. (2017a) as the main feature extractor, which is enhanced with pre-trained ViTs. In MVSFormer, ViTs work to formulate global feature correlations, while FPN is devoted to learning detailed ones. Before taking reference and source images to the ViT, we first downsample them into \((H/2, W/2)\) to save the computation and memory costs. We resize the absolute position encoding of pre-trained ViTs with bicubic interpolation to fit different image scales Dosovitskiy et al. (2020). Then plain-ViT output \(F^{(p)}\) or hierarchical-ViT output \(F^{(h)}\) is directly added to the highest level feature of the FPN encoder. Thus we can get coarse-to-fine features \(\{F^{(i)}\}_{i=1}^{l=4}\) from the FPN decoder scaled from \((H/8, W/8)\) to \((H, W)\) of the origin resolution as shown in Fig. 2(A). These features contain priors from both ViT and CNN, and will be leveraged to formulate more reliable cost volumes. We have tried other feature fusion strategies in Sec. C.2 of the Appendix, but the difference is negligible. So the simple but effective feature addition is adopted in MVSFormer.

MVSFormer-P. We use DINO as the backbone of MVSFormer-P. Furthermore, attention maps of \([CLS]\) token as in Fig. 3 are utilized to strengthen the feature learning. Since MVS is a feature matching task essentially, the priors of object or scene segmentations are useful to avoid confused depth predictions of foreground and background. As inputs to ViT have been halved yet, after the patch-wise embedding with kernel and stride size 16, DINO performs the attention-based learning for feature maps with the resolution of \((H/32, W/32)\). To better utilize the good segmentation property of DINO, we use a trainable Gated Linear Unit (GLU) Shazeer (2020) to reduce the dimension of DINO features \(\mathbf{F}_{\text{dino}}\) as shown in Fig. 2(a). We assume that \(\mathbf{A} \in \mathbb{R}^{H \times W \times h}\) indicates attention maps of the \([CLS]\) token with \(h\) attention heads from the last layer of
Thus, we repurpose learning features with multi-scale training, originated from ViT-based detection tasks. Particularly, for efficient multi-scale training, we have to ensure that 1) image sizes for each batch should be the same; 2) dynamically changing the batch size according to image sizes, which aims to make the full usage of limited memory. The PyTorch pseudo-code of this algorithm is in Alg. 1. Specifically, we train our models with dynamic resolutions from 512 to 1280, while the aspect ratios are randomly sampled from 0.8 to 0.67. Instead of the compromise by a minimum batch size, we keep the multi-scale training with the usage of limited memory. The PyTorch pseudo-code of this algorithm is in Alg. 1. Specifically, we train our models with dynamic resolutions from 512 to 1280, while the aspect ratios are randomly sampled from 0.8 to 0.67. Instead of the compromise by a minimum batch size, we keep the multi-scale training with the

2For multi-crop augmented DINO frozen in MVSFormer-P, the scale issue is alleviated; but it still limits the performance (Tab. 4).
largest batch size, assisted by the gradient accumulation. The gradient accumulation splits a batch into several sub-batches and accumulates their gradients to update the model. All instances are grouped into different pairs of resolution and sub-batch size at the start of each epoch randomly. Note that a larger image should have a smaller sub-batch to balance the memory cost and vice versa. Training with a larger batch size contributes to faster convergence with lower variances and better performance for BatchNorm layers [Ioffe 2017]. Therefore, the gradient accumulation significantly improves the efficiency of multi-scale training of MVSFormer. We find that dynamic training sizes from 512 to 1280 are sufficient to generalize MVSFormer to at least 2K resolution of Tanks-and-Templels [Knapitsch et al. 2017] as compared in Tab. 14 and Fig. 11. More details about multi-scale training are in Sec. [A] of Appendix.

Algorithm 1: PyTorch pseudo code for efficient multi-scale training with gradient accumulation

```
# B: maximal batch size
# scale_batch_dict: a mapping dict, key is resolution; value is related sub-batch size
# optimizer: Adam optimizer
for (x, y) in data_loader: # load a batch consisted of inputs x and reference depth y
    b = scale_batch_dict[shape(x)] # get related sub-batch size from the dict
    n = B // b # get the step number n for accumulation
    for i in range(n): # accumulate gradients for n steps
        optimizer.zero_grad() # clear accumulated gradients
        y_pred = MVSFormer(x[i*b:(i+1)*b]) # model inference with a sub-batch
        y = y[i*b:(i+1)*b] # select sub-batch of ground truth
        loss = LossFunction(y_pred, y) # calculate loss for sub-batch
        loss.backward() # back-propagate and accumulate gradients
    optimizer.step() # optimize model parameters
    optimizer.zero_grad() # clear accumulated gradients
```

3.3 Correlation Volume Construction

We elaborate the classical components of cost volume construction and regularization from [Gu et al. 2020; Zhang et al. 2020; Giang et al. 2021]. Note that such components are orthogonal to our key contributions, as the primary focus of this paper is on better feature extraction for MVS. To achieve the multi-stage cost volume, we first initialize a group of inverse depth ranges \{d_j\}_{j=1}^{J} for each stage. Here we omit the superscript of l-th stage for the simplification. Features of source views are warped to the reference view. Given a 2D pixel p of the reference image \(I_0\) with known camera intrinsics \(K_0, K_i\) of reference and source views, as well as their rotation \(R_{0\rightarrow i}\) and translation \(t_{0\rightarrow i}\), warped \(p_j^i\) with the j-th depth hypothesis in source image \(I_i\) can be written as

\[
p_j^i = K_i \cdot (R_{0\rightarrow i} \cdot K_0^{-1} \cdot p \cdot d_j + t_{0\rightarrow i}). \quad (3)
\]

Then the group-wise pooling [Gu et al. 2019] is leveraged to split features into \(G\) groups along the channel dimension. And the feature correlation \(C_i\) can be formulated by the inner production of group-wise reference features \(F_0^j\) and warped source features \(F_i^j\) as

\[
C_i(d_j, p, g) = \langle F_0^j(p), F_i^j(p_j^i) \rangle \in \mathbb{R}^C, \quad (4)
\]

where \(\hat{C}\) is the channel of \(F_0^j\) and \(F_i^j\). Then the feature correlation from Eq. 4 is further averaged for each group to \(C_i(d_j, p) = \frac{1}{N^2} \sum_{i=1}^{N^2} C_i(d_j, p)\) for an efficient cost volume formulation. We also train a 2D CNN to learn pixel-wise weight visibility \(\{W_i\}_{i=1}^{N^2}\) for each source view through the entropy of normalized correlations [Zhang et al. 2020; Giang et al. 2021]. Thus \(N-1\) source feature correlations can be fused with their visibility as

\[
C(d_j) = \sum_{i=1}^{N^2} W_i C_i(d_j), \quad (5)
\]

which is the input for the 3D U-Net cost volume regularization. After being regularized by the 3D U-Net, we can achieve pixel-wise output 3D cost volume \(\hat{C} \in \mathbb{R}^{D \times H \times W}\) for each stage.

3.4 Temperature-based Depth Prediction

Given the cost volume \(\hat{C}\) from the 3D U-Net, the probability volume of depth hypotheses can be achieved by the softmax along the depth dimension as \(P = \text{softmax}(\hat{C})\). REGression-based depth (REG) utilizes soft-argmin [Kendall et al. 2017] to softly weighting for each depth hypothesis, i.e., the expectation of
Figure 4: (a) Multi-stage confidence of regression and classification. Both depth ranges of regression and classification based methods miss ground truth depth (8.8) in stage-1. Regression still provides high probability for the depth upper bound. (b) An DTU example: the kettle handle can be kept by confidence map of classification-based method.

\{d_j\}_{j=1}^{D} \text{ with probability } P(d_j). \text{ For the CLAssification-based depth (CLA), the predicted depth } D_{cla} \text{ is simply selected from all depth hypotheses with the maximum probability, } i.e. \text{ the argmax depth hypothesis. Thus the regressive depth } D_{reg} \text{ and the classification depth } D_{cla} \text{ are}

D_{reg} = \sum_{j=1}^{D} d_j \cdot P(d_j), \quad D_{cla} = \arg\max_{d_j \in \{d_j\}_{j=1}^{D}} P(d_j). \quad (6)

D_{reg} \text{ is optimized with the } L_1 \text{ loss with the ground-truth depth, while } D_{cla} \text{ is optimized with the Cross-Entropy (CE) with the one-hot ground-truth depth volume.}

Remark. Peng et al. (2022) think that REGs suffer from the overfitting issue, and lead to ambiguous depth prediction, whilst CLAs are more robust but fail to achieve exact depth results. In contrast, we empirically find a different conclusion that confidence maps from CLAs are better than the REGs; and this should not be neglected especially for the widely used multi-stage MVS models Gu et al. (2020); Yang et al. (2020); Mi et al. (2021). Particularly, MVS networks can not ensure all predicted depth maps are correct, due to reflection, occlusion, or missing reliable source views. Thus, providing solid confidence maps is also important for MVS to reconstruct good point clouds as empirically evaluated in Fig. 8 and Fig. 13. However, as shown in Fig. 4(a), the REG maintains high-confidence values even for out-of-range depth hypotheses in stage-2,3,4. It is difficult for REGs to filter outliers without hurting other points of correct depth as illustrated in Fig. 4(b). Since CE can not tackle out-of-range depth labels, we advocate masking all depth outliers in training as Mi et al. (2021). Note that we also tried to optimize the MVS with masked \(L_1\) loss, but its performance is inferior to regular regression.

Although CLA has many good properties for MVS, REGs can achieve superior performance of depth and point cloud compared with CLAs in our early experiments. So we target on the issue mentioned in Peng et al. (2022), i.e., inexact depth predictions. UniMVSNet Peng et al. (2022) designs a Unified Focal Loss (UFL) to solve it, which regards CE as multiple Binary Cross-Entropy (BCE). And the focal loss Lin et al. (2017b) controlled by several hyper-parameters is used to solve the imbalance problem in BCE. Different from UFL, we propose a simple way to unify both REGs and CLAs, which only adjusts the inference process without re-training the model. We first multiply a temperature \(t\) to the cost volume \(\hat{C}\) before the softmax, and rewrite \(D_{reg}\) to the temperature-based depth expectation \(D_{tmp}\) as

\[D_{tmp} = \sum_{j=1}^{D} d_j \cdot \hat{P}(d_j), \quad \hat{P} = \text{softmax}(\hat{C} \cdot t).\quad (7)\]

Obviously, when \(t = \infty\) or \(t = 1\), \(D_{tmp}\) is equivalent to \(D_{cla}\) or \(D_{reg}\), respectively. The core idea is to adjust the temperature \(t\) during the inference to unify CLAs and REGs. For early stages with low-resolution, we set
Table 1: Quantitative point cloud results (mm) on DTU (lower is better). Best results are in bold, and second ones are underlined. * denotes that GBINet is re-tested with the same post-processing threshold to all scans for fair comparisons with other methods.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy↓</th>
<th>Completeness ↓</th>
<th>Overall↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gipuma Galliani et al. (2015)</td>
<td>0.283</td>
<td>0.873</td>
<td>0.578</td>
</tr>
<tr>
<td>COLMAP Schönberger et al. (2016)</td>
<td>0.400</td>
<td>0.664</td>
<td>0.532</td>
</tr>
<tr>
<td>R-MVSNet Yao et al. (2019)</td>
<td>0.385</td>
<td>0.459</td>
<td>0.422</td>
</tr>
<tr>
<td>AA-RMVSNet Wei et al. (2021)</td>
<td>0.376</td>
<td>0.339</td>
<td>0.357</td>
</tr>
<tr>
<td>CasMVSNet Gu et al. (2020)</td>
<td>0.325</td>
<td>0.385</td>
<td>0.355</td>
</tr>
<tr>
<td>CDS-MVSNet Giang et al. (2021)</td>
<td>0.352</td>
<td>0.280</td>
<td>0.316</td>
</tr>
<tr>
<td>UniMVSNet Peng et al. (2022)</td>
<td>0.352</td>
<td>0.278</td>
<td>0.315</td>
</tr>
<tr>
<td>TransMVSNet Ding et al. (2021)</td>
<td>0.321</td>
<td>0.289</td>
<td>0.305</td>
</tr>
<tr>
<td>GBINet* Mi et al (2021)</td>
<td>0.312</td>
<td>0.293</td>
<td>0.303</td>
</tr>
<tr>
<td>MVSFormer-P (ours)</td>
<td>0.327</td>
<td>0.265</td>
<td>0.296</td>
</tr>
<tr>
<td>MVSFormer-H (ours)</td>
<td>0.327</td>
<td>0.251</td>
<td>0.289</td>
</tr>
</tbody>
</table>

larger $t$ to make the model work as a CLA for a better global distinguishing ability. And for later stages with high-resolution, our model tends to use lower $t$ as a REG to smooth local details. In practice, we set $\{t^1, t^2, t^3, t^4\} = \{5, 2.5, 1.5, 1\}$ and achieve better performance than classification ($t = \infty$), regression ($t = 1$), and other consistent settings of $t$ as evaluated in Sec. 4.2. Note that $D_{\text{tmp}}$ is only used during testing, as the masked CLA optimized with CE is robust enough for MVS learning. Thus, $D_{\text{cla}}$ is adopted in MVSFormer for the training phase.

4 Experiments

Settings. Our methods are evaluated on DTU Aanæs et al. (2016) and Tanks-and-Temples Knapitsch et al. (2017). Since DTU data is collected in an indoor environment with fixed camera poses, our model is finetuned on the BlendedMVS dataset Yao et al. (2020) with various scenes and objects to generalize more complex environments in Tanks-and-Temples, as standard practice in Giang et al. (2021); Ding et al. (2021). MVSFormer is trained by the view number $N = 5$ of 4 coarse-to-fine stages of 32-16-8-4 depth hypotheses. CNN parts in MVSFormer are trained by Adam with a learning rate of 1e-3. DINO-small is frozen in MVSFormer-P during the training phase, while Twins-small in MVSFormer-H is trained with learning rate 3e-5 and 0.01 weights decay. Our models are trained by 10 epochs on DTU and finetuned with another 10 epochs on BlendedMVS. The learning rate is warmed up with 500 steps and then decayed with the cosine scheduler. For the multi-scale training, we dynamically change the sub-batch from 8 to 2 according to the scales from 512 to 1280, with a maximum batch size of 8. More details are listed in Sec. A of the Appendix.

4.1 Quantitative Results

DTU. Our MVSFormer is evaluated on DTU with the official evaluation metrics of point clouds, i.e, accuracy, completeness, and the overall error. The testing resolution is fixed in $1152 \times 1536$ and the view number $N = 5$. We use the depth fusion of Gipuma Galliani et al. (2015) with a consistent confidence threshold 0.5 for the point clouds. Quantitative results of DTU are shown in Tab. 1 and qualitative ones are shown in Fig. 9 and Fig. 10 of the Appendix. Note that the post-processing hyper-parameters of all scans are fixed for learning-based methods. Traditional methods Galliani et al. (2015); Schönberger et al. (2016) fail to get good completeness, which means that they have missed many points with sparse results. For learning-based methods, Our MVSFormer-H can achieve the best completeness and overall error. MVSFormer-P is the second-best in overall error which enjoys faster efficiency. Therefore, our methods can get more complete point clouds compared with other competitors. Note that results reported in GBINet Mi et al. (2021) need to use different hyper-parameters for the post-processing. Our methods can outperform GBINet with fixed

---

Such a temperature setting is always fixed for all datasets and instances without data dependency.
Table 2: Quantitative results of F-score on Tanks-and-Temples. Higher F-score means a better reconstruction quality. Best results are in bold, while the second ones are underlined.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Intermediate</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLMAP</td>
<td>42.14</td>
<td>50.41</td>
</tr>
<tr>
<td>CashMVSNet</td>
<td>56.84</td>
<td>76.37</td>
</tr>
<tr>
<td>CDS-MVSNet</td>
<td>61.58</td>
<td>78.85</td>
</tr>
<tr>
<td>TransMVSNet</td>
<td>63.52</td>
<td>80.92</td>
</tr>
<tr>
<td>UniMVSNet</td>
<td>64.36</td>
<td>81.20</td>
</tr>
<tr>
<td>GBiNet</td>
<td>61.42</td>
<td>79.77</td>
</tr>
<tr>
<td>MVSFormer-H</td>
<td>66.37</td>
<td>82.06</td>
</tr>
</tbody>
</table>

Table 3: Ablations in DTU on different pre-trained models, augmentation strategies, and loss types. Metrics are depth error ratios of 2mm ($e_2$), 4mm ($e_4$), 8mm ($e_8$) and Overall error (Ovl.) of point clouds. Red line denotes our baseline results; green and blue rows refer to our MVSFormer-P and MVSFormer-H respectively.

<table>
<thead>
<tr>
<th>Pre-trained</th>
<th>Augmentation</th>
<th>Loss</th>
<th>$e_2$ ↓</th>
<th>$e_4$ ↓</th>
<th>$e_8$ ↓</th>
<th>Ovl. ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cropping</td>
<td>HR-FT</td>
<td>Multi-scale</td>
<td>REG</td>
<td>CLA</td>
<td>22.81</td>
</tr>
<tr>
<td>ResNet50</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>20.09</td>
</tr>
<tr>
<td>ResNet50</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>20.38</td>
</tr>
<tr>
<td>DINO-small</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>22.06</td>
</tr>
<tr>
<td>DINO-small</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>17.06</td>
</tr>
<tr>
<td>DINO-small</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>20.98</td>
</tr>
<tr>
<td>DINO-small</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>17.18</td>
</tr>
<tr>
<td>MAE-base</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>18.45</td>
</tr>
<tr>
<td>Twins-small</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>22.62</td>
</tr>
<tr>
<td>Twins-small</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>17.70</td>
</tr>
<tr>
<td>Twins-small</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>17.50</td>
</tr>
</tbody>
</table>

hyper-parameter settings for all scans. With the impressive improvements achieved by MVSFormer, we think that pre-trained ViTs have the potential to push the limits of MVS.

**Tanks-and-Temples.** Our anonymous submission of MVSFormer-H has ranked Top-1 on both intermediate and advanced sets of the official Tanks-and-Temples leaderboard compared with other published works since May/2022. We show quantitative results on both intermediate set and advanced set in Tab. 14. All instances are inferred with the original 1088 × 1920 image size, and more detailed settings are in the Appendix. The metric is officially evaluated by the F-score based on precision and recall of submitted point clouds Knapitsch et al. (2017). MVSFormer-H outperforms all other state-of-the-art methods with mean F-scores of 66.37 and 40.87 for intermediate and advanced sets respectively. As shown in Tab. 14, our method can achieve the best or second-best results in almost all cases except ‘Auditorium’, which demonstrates its good generalization and impressive performance. Furthermore, confidence maps from CLA can filter outliers and get more precise point clouds as shown in the Appendix. Besides, the proposed multi-scale strategy can generalize MVSFormer to fit larger resolutions, such as 2K. More qualitative results are in Fig. 11 and Fig. 12 of the Appendix.

### 4.2 Ablation studies

**Different pre-trained models.** We have tested different pre-trained models for MVS in Tab. 3 which include ResNet50 [He et al. (2016)], DINO [Caron et al. (2021)], MAE [He et al. (2021)], and Twins [Chu et al. (2021a)]. Our baseline method is a 4-stage MVS model with visibility modules [Zhang et al. (2020)] and the random cropping [Mi et al. (2021)]. From Tab. 3, ResNet50 in row2 improves the depth but fails to reduce the overall error of point clouds. Because CNN-based pre-training can not learn proper features from reflection and texture-less areas, which causes discouraged metrics for these scans and leads to even worse results in point clouds as shown in Fig. 5. Our MVSFormers (row4 and row9 of Tab. 3) can improve the baseline even without the multi-scale training. For the comparisons among pre-trained ViTs with the multi-scale training strategy, both DINO-small and MAE-base are frozen during the training. But DINO-small achieves better
performance with fewer parameters, which is benefited by the multi-crop and the proposed attention based GLU. Although REG based methods enjoy slightly better depth, they fail to reconstruct proper point clouds with proper confidence. MVSFormer-H and -P achieve the best results in point cloud and depth respectively, while the pre-trained ResNet50 based model with all other tricks (row3 of Tab. 3) fails to produce competitive results. Detailed comparison among MVSFormers and ResNet are discussed in Sec. C.5. The computation and GPU memory cost of different pre-trained MVS methods are discussed in Sec. B.

Multi-scale training. From Tab. 3, both MVSFormer-P and -H achieve considerable improvements from the multi-scale training. Especially, the trainable Twins enjoys much more benefits (0.19 in the overall error). Because of the spatial invariance, CNN-based ResNet50 basically can not be benefited from multi-scale training (row3). Besides, we also compare multi-scale strategy with high-resolution finetuning (HR-FT), i.e., finetuning the model with fixed 1024×1280 images for another 5 epochs. But HR-FT produce inferior results compared with the multi-scale one. We think that finetuning on a specific resolution still suffers from the spatial overfitting to attention blocks, which limits the generalization of ViTs for various image scales.

The effect of pre-training for ViTs. Although training a transformer (with CNN) for MVS is feasible Zhu et al. (2021), we think that the pre-training is still important for MVS learning, especially for the feature learning in our work. In particular, we train our MVSFormer-H with Twins-small from the scratch as shown in Tab. 4(left). We increase the learning rate of no pre-trained Twins-small from 3e-5 to 1e-4, while all other settings are unchanged. Results from Tab. 4(left) show that pre-training is critical to our proposed MVSFormer. The Twins-small without pre-training is not as good as pre-trained one. Without the pre-training, our methods are close to those attention-based MVS methods Ding et al. (2021); Zhu et al. (2021) with only intra-view attention except the temperature based depth. Actually, no pre-trained MVSFormer-H performs similarly as TransMVSNet Ding et al. (2021). So the pre-training is important for ViTs to model proper feature representations to tackle the essential feature matching problem in MVS.

Effects of DINO attention maps from \( \text{CLS} \) and GLU. In Tab. 4(right), we evaluate the performance of MVSFormer-P with and without the \( \text{CLS} \) attention map. Furthermore, we compare GLU based attention map fusion with the simple concatenation and \( \times 2 \) convolutions to balance the parameters. From Tab. 4(right), both attention fusion strategies enjoy improvements compared with the baseline without such attention map. And the GLU block can achieve better depth with the same computation.

REG vs CLA in MVS. To show the importance of effect confidence map from CLA models during point cloud reconstruction. In Fig. 6 we further show the difference between the REG and CLA on the real-world Tanks-and-Temples reconstruction. Note that CLA can produce more reliable confidence maps to distinguish certain foregrounds and uncertain backgrounds or sky. Thus CLA based methods reconstruct more valid
Figure 6: ‘Francis’ and ‘Horse’ in Tanks-and-Temples compared between REG and CLA based MVSFormer-H. The number of total points are listed in the brackets of (b).

Temperature-based depth prediction. Quantitative ablations about REG and CLA of MVSFormer-H are shown in Tab. 5. The vanilla CLA ($t = \infty$) based model can not achieve better results compared with the one trained with REG. As reducing the temperature $t$, depth is smoothed and models tend to reconstruct more accurate point clouds. Because of the decrease in accuracy distance and the increase in completeness distance. Since the trade-off of the accuracy and completeness, $t = 0.75$ achieves a worse overall result. Simply reducing $t$ for early stages causes over-smoothing results, which harms the completeness of point clouds. Our $\{t^1, t^2, t^3, t^4\} = \{5, 2.5, 1.5, 1\}$ setting can get a good trade-off in both depth and point clouds, which is better than any consistent $t$; it also enjoys better accuracy compared with REG, which indicates that CLA provides better confidence maps to filter outliers. Therefore, the idea of making early stages work as CLA and latter stages work as REG is reasonable. We have also tested other settings following the decreasing temperatures, but the difference is not obvious. More concrete quantitative analysis are discussed in Sec. C.6.

5 Conclusion

In this paper, we discuss the influence of pre-trained models on MVS learning, and propose a ViT enhanced MVS architecture called MVSFormer. MVSFormer achieves prominent improvements with a better feature encoding enhanced by pre-trained ViTs. Furthermore, MVSFormer can be divided into MVSFormer-P and MVSFormer-H with frozen plain-ViTs and trainable hierarchical-ViTs. The efficient multi-scale training is used to generalize MVSFormer to various resolutions. Besides, an effective temperature-based depth prediction is proposed to unify both REG and CLA in the MVS learning, which produces smooth depth maps and clear point clouds without outliers. Our method can achieve state-of-the-art results in DTU, and rank top-1 on Tanks-and-Temples.
References


Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.


A More Implemented Details and Discussions

**Multi-scale training.** The training scales are randomly selected from 25 patterns, whose height is ranged from 512 to 1024, while width is ranged from 640 to 1280. We also randomly crop images from [0.83,1.0] of the original scale. The global batch size is 8, while the relation of sub-batch size and resolution is shown in Tab. 6. Thanks for the mixed-precision, it only takes about 22 and 15 hours for our proposed MVSFormer to be trained with 10 epochs in DTU Aanæs et al. (2016) and BlendedMVS Yao et al. (2020) respectively with two V100 32GB NVIDIA Tesla GPUs.

Table 6: The relation of sub-batch size and resolution in the multi-scale training of MVSFormer-P and MVSFormer-H, which can be trained in two 32GB GPUs with maximum batch size 8.

<table>
<thead>
<tr>
<th>Resolution</th>
<th>sub-batch size</th>
<th>MVSFormer-P</th>
<th>MVSFormer-H</th>
</tr>
</thead>
<tbody>
<tr>
<td>512x640–768</td>
<td>8</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>576x704–832</td>
<td>8</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>640x832–960</td>
<td>8</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>704x896–1024</td>
<td>8</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>768x960–1088</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>896x1152–1280</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>960x1216–1344</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1024x1280</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

**Data augmentation.** We augment input images for ViT in MVSFormer with the same parameter for all reference and source views, which can slightly improve the depth performance as shown in Tab. 7. Specifically, the data augmentation is performed by random gamma, color jitters with brightness, contrast, saturation, and hue. The augmented parameters are sampled from uniform distributions in the ranges [0.9,1.1] for gamma, contrast and saturation, [0.8,1.2] for brightness, [0.95,1.05] for hue.

Table 7: Depth metrics on DTU (512 × 640) with error ratios of 2mm (e_2), 4mm (e_4), 8mm (e_8), and 14mm (e_{14}) with/without data augmentation.

<table>
<thead>
<tr>
<th></th>
<th>e_2</th>
<th>e_4</th>
<th>e_8</th>
<th>e_{14}</th>
</tr>
</thead>
<tbody>
<tr>
<td>w./o. aug</td>
<td>17.15</td>
<td>10.85</td>
<td>7.29</td>
<td>5.38</td>
</tr>
<tr>
<td>with aug</td>
<td><strong>16.52</strong></td>
<td><strong>10.46</strong></td>
<td><strong>7.07</strong></td>
<td><strong>5.22</strong></td>
</tr>
</tbody>
</table>
Post-processing. For DTU, we get the final point clouds with the depth fusion tool from Gipuma et al. (2015) with consistent hyper-parameters, i.e., disparity threshold 0.1, number consistent 2, and probability threshold 0.5. For Tanks-and-Temples, to avoid adjusting hyper-parameters for each case, we follow the dynamic consistency checking proposed in Yan et al. (2020).

Why completely separate the FPN and ViT in MVSFormer? The FPN used in MVS can efficiently integrate the information from different input scales. And FPN is not redundant to our ViT in MVSFormer for three reasons. 1) We use 1/2 input scale for ViT, while FPN takes full resolution inputs. So FPN can learn more detailed information. In Sec. C.1 we have discussed the performance of ViTs without FPNs and pre-trained ResNet. ViTs can work complementarily with FPNs for both global understanding and local details in MVS. 2) Since the inputs to ViT are downsampled to 1/2 to balance the computation, the valid largest feature scale of Hierarchical ViT is 1/8, while the one of Plain ViT is smaller (1/32). Our feature scales of FPN are 1, 1/2, 1/4, 1/8 respectively. So only one feature layer (1/8) can be completely replaced with the ViT feature. For the reason of the model integrity, we remain all layers of the FPN. 3) Because the inputs to FPN are 2x larger than ones to ViT, the highest level feature (1/8) of FPN also contains informative features learned from high-resolution inputs which are not included in ViTs.

Why using DINO attention maps from [CLS] rather than other tokens? We use the attention map attached from the [CLS] token for two reasons. 1) DINO Caron et al. (2021) is pre-trained with self-distillation. The [CLS] token is directly supervised during the pre-training. So the [CLS] token needs to understand the global semantics in the image. Thus the attention map of the [CLS] token is more significant to decide the final prediction. 2) Moreover, the attention map from all tokens (HW × HW) is much larger than ones from [CLS] token (H × W). Thus choosing or getting a reasonable attention map from HW × HW relations is also a difficult issue.

B Inference Memory and Time Costs

We test the inference memory and time costs with input resolution 1152 × 1536 in Tab. 8 compared with CNN-based pre-training and pure FPN. All comparisons are based on a V100 NVIDIA Tesla GPU. From Tab. 8 although the parameter scale is increased, ViT enhanced MVSFormers only cost a little more GPU memory and time compared with the baseline method. Except for parameters contained by ViT itself, most other trainable parameters are worked for the dimension reduction in MVSFormer. MVSFormer-H can infer faster compared with MVSFormer-P, which is benefited from the efficient multi-scale attention designed in Twins Chu et al. (2021a). Note that the pre-trained CNN model –ResNet50 also costs a lot in GPU memory and inference time.

<table>
<thead>
<tr>
<th></th>
<th>Memory (MB)</th>
<th>Time (s/img)</th>
<th>Params. (all)</th>
<th>Params. (trainable)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>4262(8997)</td>
<td>0.4125(0.4055)</td>
<td>1.35M</td>
<td>1.35M</td>
</tr>
<tr>
<td>Baseline+ResNet50</td>
<td>4882(9523)</td>
<td>0.4920(0.4704)</td>
<td>37.27M</td>
<td>37.27M</td>
</tr>
<tr>
<td>MVSFormer-P</td>
<td>4842(9251)</td>
<td>0.5530(0.5230)</td>
<td>26.45M</td>
<td>4.79M</td>
</tr>
<tr>
<td>MVSFormer-H</td>
<td>4970(9431)</td>
<td>0.4831(0.4398)</td>
<td>28.01M</td>
<td>28.01M</td>
</tr>
</tbody>
</table>

C More Experiment Results

C.1 Pre-trained ViTs without FPNs

To explore the learning ability of ViTs, some quantitative results about ViTs (DINO-small Caron et al. (2021), MAE-base He et al. (2021), Twins-small Chu et al. (2021a)) trained without FPNs for MVS are shown in Tab. 9 which are also compared with CNN-based pre-trained ResNet34 and ResNet50 and the vanilla FPN. Note that these experiments are only based on low-resolution cases in DTU of 256 × 320, because we want
to save computation with all trainable ViT weights. The learning rates of MAE and DINO are 1e-5; the learning rate of Twins is 3e-5, and the ones of all CNNs are 1e-3. From Tab.9 ViTs cannot achieve as good details as CNNs (2mm, 4mm), but results from ViTs are more robust in large depth error metrics (8mm, 14mm). Therefore, we think that ViTs can tackle some serious mistakes caused by reflection and texture-less areas as mentioned in the main paper. Interestingly, FPN trained from scratch achieves better depth results in low-resolution cases compared with other pre-trained CNNs, which demonstrates the dilemma of using pre-trained CNNs in MVS again. Twins-small can get better depth than FPN except for the 2mm error benefited by its pyramid architecture. So ViTs can work complementarily with CNN-based FPNs for both global understanding and local details in MVS.

Table 9: Depth metrics for DTU (256 × 320) compared with FPN, pre-trained CNNs and ViTs He et al. (2021); Caron et al. (2021); Chu et al. (2021a). Note that FPN is trained from scratch without pre-training.

<table>
<thead>
<tr>
<th>Error(%)</th>
<th>FPN</th>
<th>ResNet34</th>
<th>ResNet50</th>
<th>MAE-base</th>
<th>DINO-small</th>
<th>Twins-small</th>
</tr>
</thead>
<tbody>
<tr>
<td>2mm</td>
<td>19.53</td>
<td>20.65</td>
<td>20.55</td>
<td>21.85</td>
<td>26.66</td>
<td>19.65</td>
</tr>
<tr>
<td>4mm</td>
<td>11.85</td>
<td>12.43</td>
<td>12.32</td>
<td>12.42</td>
<td>14.42</td>
<td>11.53</td>
</tr>
<tr>
<td>8mm</td>
<td>8.25</td>
<td>8.55</td>
<td>8.45</td>
<td>8.17</td>
<td>8.87</td>
<td>7.92</td>
</tr>
<tr>
<td>14mm</td>
<td>6.48</td>
<td>6.75</td>
<td>6.60</td>
<td>6.33</td>
<td>6.60</td>
<td>6.24</td>
</tr>
<tr>
<td>mean</td>
<td>11.52</td>
<td>12.09</td>
<td>11.98</td>
<td>12.19</td>
<td>14.14</td>
<td>11.34</td>
</tr>
</tbody>
</table>

C.2 Different Feature Fusion Strategies of MVSFormer

We pay more attention to the essential improvements gained from pre-trained ViTs in this paper. Thus we tend to use simple feature fusion strategies in MVSFormer. Both the Direct Feature Addition (DFA) (used in the main paper) and the Multi-scale Feature Addition (MFA) is considered in Tab.10 For the multi-scale addition, we use extra convolution blocks to further upsample ViT features to 1/4 and 1/2, and add them to related feature maps in FPN. Since inputs to ViTs are halved, we do not try to upsample ViT features to 1/1. From Tab.10 MFA can achieve slightly better depth predictions but worse point cloud metrics. We think that ViT features are not suitable for high-resolution features, and adopt DFA as our solution.

Table 10: Ablations about different feature fusion strategies of Direct Feature Addition (DFA) and Multi-scale Feature Addition (MFA) in MVSFormer-H.

<table>
<thead>
<tr>
<th>$e_2$</th>
<th>$e_4$</th>
<th>$e_8$</th>
<th>Acc.</th>
<th>Cop.</th>
<th>Ovl.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFA</td>
<td>17.50</td>
<td>12.48</td>
<td>9.14</td>
<td>0.327</td>
<td>0.251</td>
</tr>
<tr>
<td>MFA</td>
<td>17.53</td>
<td>12.24</td>
<td>8.62</td>
<td>0.329</td>
<td>0.253</td>
</tr>
</tbody>
</table>

C.3 Qualitative Results of Temperature-based Depth Prediction

We show additional qualitative results of CLA with different temperature $t$ in Fig.7 $t = 100$ tends to output depth maps with jagged boundaries, while depth maps with $t = 1$ and based on regression suffer from uncertain and ambiguous predictions. Although the visual difference is not obvious between Fig.7(c) and Fig.7(f), our setting of $\{t_1, t_2, t_3, t_4 = 5, 2.5, 1.5, 1\}$ can get more exact depth predictions compared with $t = 100$ or $t = \infty$, which leads to better detailed depth and point clouds results as discussed in the main paper.

C.4 Comparison of Different ViT Capacities

We further evaluate the performance of MVSFormer-P and MVSFormer-H with larger ViT backbones (base model) in Tab.11 Interestingly, DINO-base achieves worse performance compared with DINO-small. We think that smaller DINO Caron et al. (2021) used in MVSFormer-P enjoys better generalization, because the DINO backbone is fixed in MVSFormer-P due to the costly plain-ViT design. On the other hand, Twins-base can achieve full depth improvements compared with the small one, but improvements of point clouds are
negligible. Obviously, point cloud metrics are more difficult to be improved. But we can still expect for good performance from larger trainable pre-trained ViTs. Besides, both DINO-base and Twins-base make training be converged faster compared with small ones.

### C.5 The Performance of Pre-trained ResNet50 with All Other Techniques

To further ensure the effectiveness of pre-trained ViTs for MVS, we provide more results in Tab. 12 about pre-trained ResNet50 with all other techniques used in our MVSFormer including the multi-scale training. Since CNNs enjoy good spatial invariance, the multi-scale training for ResNet50 is not as important as one for ViT based MVSFormers. The external experiment in Tab. 12 shows that multi-scale training can not improve ResNet50 a lot.

Table 12: Results of ResNet50 with all other proposed components. ‘T-CLA’ indicates temperature based depth with CLA.

<table>
<thead>
<tr>
<th>Pre-trained</th>
<th>Multi-scale</th>
<th>T-CLA</th>
<th>$e_2 \downarrow$</th>
<th>$e_4 \downarrow$</th>
<th>$e_8 \downarrow$</th>
<th>Acc.↓</th>
<th>Cop.↓</th>
<th>Ovl.↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet50</td>
<td>✓</td>
<td>✓</td>
<td>20.09</td>
<td>15.11</td>
<td>11.78</td>
<td>0.323</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ResNet50</td>
<td>✓</td>
<td>✓</td>
<td>20.38</td>
<td>13.87</td>
<td>10.37</td>
<td>0.312</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DINO-small</td>
<td>✓</td>
<td>✓</td>
<td>17.18</td>
<td>11.96</td>
<td>8.53</td>
<td>0.296</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Twins-small</td>
<td>✓</td>
<td>✓</td>
<td>17.50</td>
<td>12.48</td>
<td>9.14</td>
<td>0.289</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### C.6 Quantitative Analysis of REG, CLA, and Temperature-based CLA

Here we re-list these important results in Tab. 13 from Tab. 5 for a clear and concrete illustration. The Accuracy (Acc) of point clouds indicates the mean value of reconstructed points whose closest distance to the ground-truth points. So ‘Acc’ can be seen as the precision of reconstructed points from the model. On the other hand, ‘Cop’ shows how complete the proposed method reconstructs. From Tab. 13, compared with
REG, our temperature based depth from CLA can achieve better Acc and Ovl, with similar Cop, which means that our CLA enjoys better confidence maps that can filter outliers near the boundaries successfully with more precise point clouds. Note that vanilla CLA fails to get good Acc (0.349 vs 0.336) with similar Cop (0.248 vs 0.249). It means that the points reconstructed by inexact argmax from vanilla CLA are not precise (vanilla CLA does not produce more points because Cop are similar), which can be greatly solved by our temperature based depth.

Table 13: Quantitative point cloud results of REG, CLA, and our temperature-based CLA of MVSFormer-H.

<table>
<thead>
<tr>
<th></th>
<th>Accuracy (Acc) mm</th>
<th>Completeness (Cop) mm</th>
<th>Overall (Ovl) mm</th>
</tr>
</thead>
<tbody>
<tr>
<td>REG</td>
<td>0.336</td>
<td>0.249</td>
<td>0.293</td>
</tr>
<tr>
<td>vanilla CLA</td>
<td>0.349</td>
<td>0.248</td>
<td>0.298</td>
</tr>
<tr>
<td>ours</td>
<td>0.327(-0.009)</td>
<td>0.251(+0.002)</td>
<td>0.289(-0.004)</td>
</tr>
</tbody>
</table>

C.7 Ablations about More Source Views

Figure 8: Qualitative ablations about the view number. All visualized depths are filtered by the confidence map > 0.4.

During the testing on Tanks-and-Temples [Knapitsch et al. 2017], we find that camera poses from the advanced set are very challenging. Therefore we try to expand source views with candidates of the selected source views which have not been included according to the view selection from [Yao et al. 2018]. We find that increasing the number of source views can effectively improve the performance in complex scenes of Tanks-and-Temples. Thanks to the visibility normalization [Zhang et al. 2020; Giang et al. 2021] used in MVSFormer, our method can be generalized to $N = 15, 20$, and achieve better results on Tanks-and-Temples. As shown in Fig. 8, depth maps predicted by 20-view inputs are more reliable than ones from 10-view. Quantitative results from Tab. 14 show that the state-of-the-art results on Tanks-and-Temples from our main paper can be further improved with more source views.

Table 14: Ablation studies on the number of total views (reference and source views).

C.8 Qualitative Results of DTU

We provide qualitative results of DTU compared with CDS-MVSNet [Giang et al. 2021] and GBiNet [Mi et al. 2021]. Qualitative depth and confidence comparisons are shown in Fig. 9, while point clouds are compared in Fig. 10. From Fig. 9, our MVSFormer-H can achieve more robust depth maps compared with others. Notably,
depth maps from MVSFormer-H are as smooth as the regressive depth got from CDS-MVSNet. Besides, the \textit{argmax} operation used in GBiNet fails to achieve stable depth predictions, and relies heavily on confidence maps to filter invalid depth. But our MVSFormer can get not only good depth predictions but also reliable confidence maps, which is benefited from the proposed temperature-based depth prediction. From Fig. 10, our method can faithfully reconstruct some challenging point clouds, which are usually omitted by other competitors.

![Figure 9: Qualitative DTU depth and confidence compared with CDS-MVSNet Giang et al. (2021), GBiNet Mi et al. (2021), and our MVSFormer-H.](image)

![Figure 10: Qualitative DTU point clouds compared with CDS-MVSNet Giang et al. (2021), GBiNet Mi et al. (2021), and our MVSFormer-H. Please zoom-in for details.](image)

**C.9 Qualitative Results of Tanks-and-Temples**

Our qualitative depth results of Tanks-and-Temples are shown in Fig. 11. Benefited by the proposed temperature-based depth prediction, MVSFormer-H can achieve not only good depth predictions but also reliable confidence maps, which leads to high-quality filtered depth in Fig. 11(d).
We also provide qualitative results of Tanks-and-Temples compared with TransMVSNet [Ding et al. (2021)] and UniMVSNet [Peng et al. (2022)]. Fig. 12 shows qualitative results of ‘Horse’ and ‘Lighthouse’ in the Tanks-and-Temples intermediate set. From Fig. 12, our MVSFormer-H can reconstruct more details (better Recall) and generate point clouds with more accurate positions (better Precision). But TransMVSNet misses some structures in ‘Horse’ and predicts biased points in ‘Lighthouse’. On the other hands, UniMVSNet suffers from many outliers in ‘Horse’, and fails to reconstruct a correct lighthouse.

Figure 11: Depth prediction, depth confidence, and filtered depth of our MVSFormer-H on Tanks-and-Temples. Depth maps are filtered by confidence > 0.5 in (d).

D Real-World Results

We show some challenging real-world cases in Fig. 13 compared with CasMVSNet [Gu et al. (2020)]. Colmap [Schonberger et al. (2016)] is utilized for camera poses. We use the same setting of Gipuma fusion [Galliani et al. (2015)] with disparity threshold 0.2, number consistent 3 for both CasMVSNet and MVSFormer-H. The confidence thresholds are set as 0.9 and 0.5 for CasMVSNet and MVSFormer-H respectively, which follow their DTU settings. From Fig. 13, CasMVSNet suffers from messy outliers without segmentation masks, while our MVSFormer-H can achieve good results. Besides, our method can get robust results for texture-less and reflective objects.

E More Point Cloud Results

We show all point clouds of DTU test set generated by our MVSFormer-H in Fig. 14. And point clouds of both intermediate and advanced sets of Tanks-and-Temples are shown in Fig. 15.

F Limitations and Future Works

We discuss the limitations and potential future works. In particular, 1) we utilize the recent ViTs pre-trained with self-supervised (MAE [He et al. (2021)], DINO [Caron et al. (2021)]) and supervised (Twins [Chu et al. (2021a)]) tasks, while it is an interesting future work of exploring the influence of different pre-training tasks on MVS. The involved methods – MAE, DINO, and Twins are relatively representative. Specifically, MAE
<table>
<thead>
<tr>
<th>Method</th>
<th>Horse</th>
<th>Lighthouse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>59.47</td>
<td>65.03</td>
</tr>
<tr>
<td>TransMVSnet</td>
<td>56.94</td>
<td>62.54</td>
</tr>
<tr>
<td>UniMVSnet</td>
<td>53.11</td>
<td>63.46</td>
</tr>
</tbody>
</table>

Figure 12: Qualitative results of Tanks-and-Temples (Horse and Lighthouse), compared with TransMVSNet [Ding et al. 2021], UniMVSNet [Peng et al. 2022], and our MVSFormer-H. \( \tau \) is the distance threshold provided officially. \( \tau = 3 \) mm and \( \tau = 5 \) mm for ‘Horse’ and ‘Lighthouse’ respectively.

Figure 13: Point Clouds of two real-world cases without masking compared with CasMVSNet [Gu et al. 2020]. These cases are challenging in complex structures (the first row), hairy, texture-less and reflective objects (the second row).

and DINO are based on the vanilla plain-ViT, while Twins is based on the hierarchical-ViT. Additionally, note that due to the memory cost limitation, we have to fix ViT weights in MVSFormer-P, which loses some advantages over another variant – MVSFormer-H. Therefore, these ViTs are not ensured to be pre-trained with the same architecture and training settings in our paper. So it is interesting to use the same ViT architecture for different pre-training tasks, and further explore how these pre-training tasks (supervised and self-supervised) affect MVS. However, re-training all of these ViTs with different pre-training tasks is very
Figure 14: Point Clouds of all test set in DTU Aanæs et al. (2016) reconstructed by MVSFormer-H.
Figure 15: Point Clouds of Tanks-and-Temples [Knapitsch et al. 2017] reconstructed by MVSFormer-H.
non-trivial, and demands extraordinary computing resources. Thus we take it as future work. Critically, the performance of our proposed models has already outperformed all existing methods; this demonstrates the efficacy of our models. 2) Despite being simple, the fusion method used in this paper is good enough to make our models competitive. Here, we only consider the single and multi-scale feature addition. On the other hand, more technical and informative fusion strategies would be much advisable, such as cross-attention [Vaswani et al. (2017)] and GRU modules [Cho et al. (2014)]. This however would also be a very interesting future work that may inspire the community.

G Broader Impact

Our methods can take the 3D reconstruction based on 2D images. Since learning-based MVS methods can be generalized to various real-world datasets, the proposed method may cause some societal impacts with controversial 2D images. Note that we only provide technical methods in this paper, but the real-world practice with potential negative societal impacts should be further considered.