GeoMVSNet: Learning Multi-View Stereo with Geometry Perception Supplementary Material

Zhe Zhang[1] Rui Peng[1] Yuxi Hu[2] Ronggang Wang[1]
[1]School of Electronic and Computer Engineering, Peking University, China
[2]School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China
doublez@stu.pku.edu.cn rgwang@pkusz.edu.cn

In the supplementary material, we start with detailing the full-scene depth distribution of MVS scenarios under the GMM assumption in Sec. I. In Sec. II, we review the cascade-based MVS framework and provide details of the geometry fusion network and the geometry embedding. In Sec. III, we present our fusion strategy, showing additional qualitative reconstruction results by our model, comparing the computational effectiveness of recent methods, and giving more ablation studies. And we discuss the limitation of the proposed GeoMVSNet in Sec. IV.

I. Depth Distribution of MVS Scenarios

In this section, we describe in more detail how to model the full-scene depth distribution based on the Gaussian-Mixture Model. Different MVS scenarios generally have different external properties, *e.g.* total number of viewpoints, scene depth extremes, etc. Therefore, it is unreliable to reflect the overall distribution of the whole scene by only counting the depth values of a single view.

In contrast, we count the depth values of all viewpoints on valid pixels for each scene. The 2D plane view of Fig. 5 of the main text is shown in Fig. 9. Based on the plane sweeping algorithm [66], we usually divide the pre-estimated depth range into M depth hypothesis planes, and we set $M = 64$ bins for calculating the distribution histograms for statistical depth values of all pixels among all viewpoints. And we use the GMM with $K = 1$ and $K = 2$ to fit the full-scene depth distribution.

Most scenarios can be well portrayed at $K \leq 2$, but Fig. 8 shows the special case. From the figure, we can see that the depth value distribution of the scene has three distinct peaks. However, the GMM assumption does not fail, and we can still compute the similarity of depth values belonging to each bin in a discrete way as mathematically expressed by Equ. 11 and Equ. 12 in the main text.

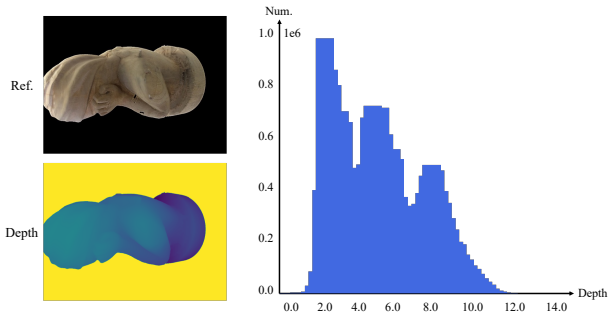

Figure 8. **Special case of MVS scenario with $K = 3$ under the GMM assumption on the BlendedMVS dataset.**

II. Supplement of Methodology

In this section, we first review the pipeline of the learning-based MVS network and the cascade-based MVS framework in Sec. II.1. Then, we provide the specific data structures and parameters of the proposed two-branch geometry fusion network in Sec.II.2 and show more examples of the probability volume geometry embedding in Sec. II.3.

II.1. Review of Cascade-based MVS Framework

Most end-to-end MVS frameworks follow the classic pipeline of MVSNet [70]. The deep feature $\{F_i\}_{i=0}^{N} \in \mathbb{R}^{C\times H\times W}$ are firstly extracted from the input reference image I_0 and source images $\{I_i\}_{i=1}^{N}$. Different feature channels represent different descriptions and portrayals of the input RGB images. Afterward, the differentiable homography is used to warp source features to the reference camera frustum by

$$H_{i\to 0}(d) = dK_i T_i T_0^{-1} K_0^{-1} , \tag{14}$$

where K and $T = \{R \mid t\}$ refer to the camera intrinsics and extrinsics respectively, and d is sampled from $[d_{min}, d_{max}]$.

We adopt the group-wise correlation [67, 68] strategy to reduce the channel dimension. Let $F_0(z)^g$ and $F_i(z)^g$ be

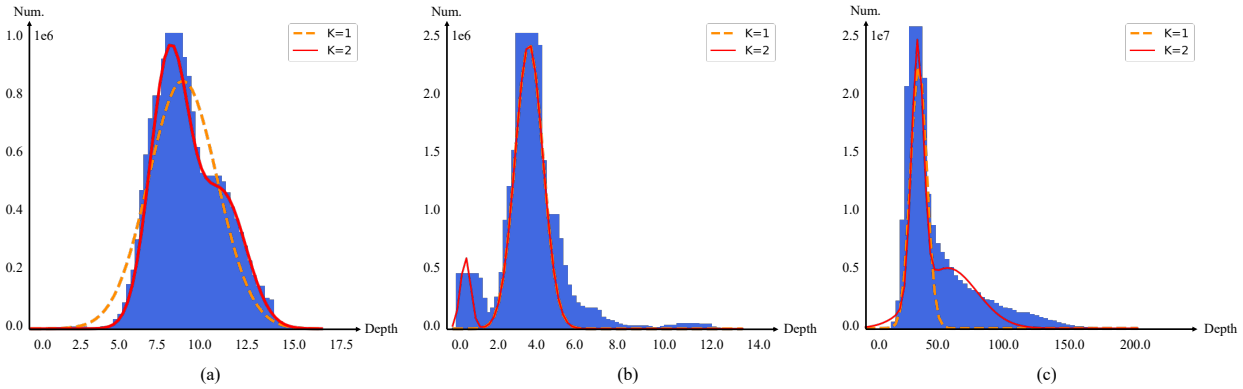

Figure 9. **Plan view of depth distributions of Fig. 5 in the main text.**

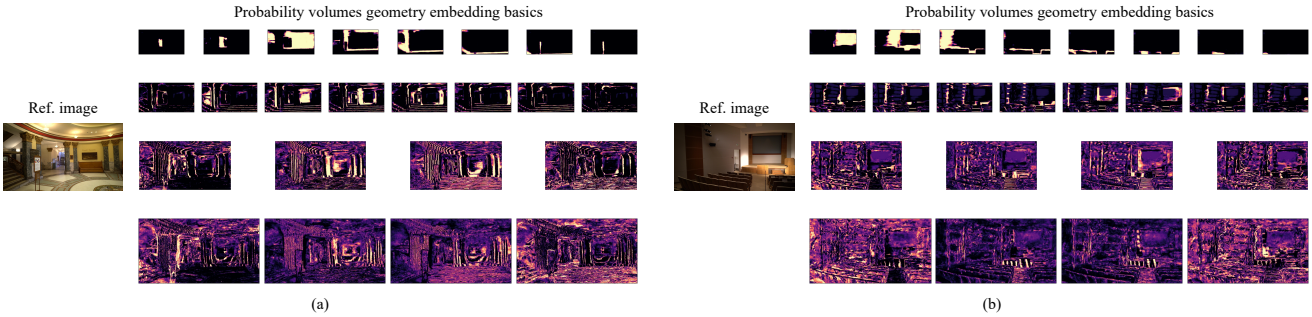

Figure 10. **More examples of the probability volume geometry embedding on the advanced set of Tanks and Temples dataset.** (a) Scene of the Museum; (b) scene of the Auditorium.

the g-th group feature of F_0 and F_i, the g-th group similarity is computed as

$$S_i(z)^g = \frac{G}{C} \left\langle F_0(z)^g, F_i(z)^g \right\rangle \ , \tag{15}$$

where $\langle :, : \rangle$ denotes the inner product, and the respective group similarity vector $S_i \in \mathbb{R}^{G \times M \times H \times W}$. To handle an arbitrary number of input viewpoints, the aggregation process is applied for assembling the cost volume $C \in \mathbb{R}^{G \times M \times H \times W}$. And the adaptive aggregation [71] is used to re-weight the contribution of pixels,

$$C = \frac{\sum_{i=1}^{N-1} w_i S_i}{\sum_{i=1}^{N-1} w_i} \ . \tag{16}$$

The cost volume encodes the cost matching between the reference image and all paired source images under spatial division. And learning-based MVS methods further use neural networks to optimize the rough and incorrect matching and generate the probability volume $P \in \mathbb{R}^{M \times H \times W}$. As for the final depth estimation, we adopt the classification approach and generate the depth map D from P by applying the winner-takes-all [66].

In an extension of the above single-stage process, the cascade-based architecture uses early depths to narrow the depth hypothesis in a coarse-to-fine manner. And we explicitly integrate the geometric priors from coarse stages into finer stages to exploit the full-scene geometry perception.

II.2. Geometry Fusion Network

Details of the specific data structures and parameters of the geometry fusion network are shown in Tab. 6. We build the two-branch submodules $\mathcal{B}$ and $\hat{\mathcal{B}}$ using the "Conv." block, "ResBlock*" block, and "DeResBlock*" block. And the *Fusion* network integrates the structural features with the original FPN feature among different stages. The geometric priors from coarse stages are explicitly encoded into the feature extraction process in finer stages, laying a solid foundation for robust aggregation.

II.3. Geometry Embedding

We present the geometry embedded in the circular vault structure in Fig. 3 of the main text. And Fig. 10 visualizes more examples of the probability volumes geometry embedding of planar areas and cylindrical man-made structures. We can see that geometric clues of the scene are re-

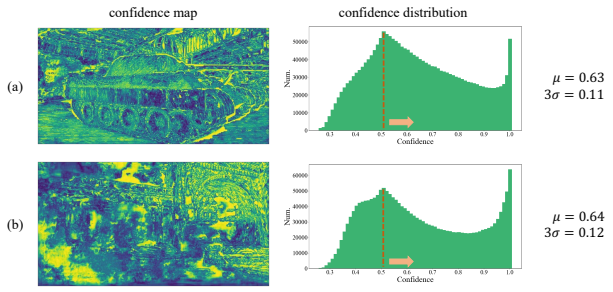

Figure 11. **Confidence distribution on the Tanks and Temples dataset.** (a) Scene of the Panther; (b) scene of the Temple. Darker means greater confidence produced by the network.

Figure 12. **Visualization of the reconstruction point clouds of the aerial photography on the BlendedMVS dataset.**

covered continuously and finely while they are embedded into the cost regularization network at finer stages, helping the network to learn full-scene geometry perception better.

III. Supplement of Experiments

In this section, we describe our depth map fusion strategy in more detail in Sec. III.1 and give more qualitative point cloud reconstruction results in Sec. III.2. Then, we discuss the computational effectiveness of run-time and GPU consumption in Sec. III.3 and give more ablation studies of our model in Sec. III.4.

III.1. Depth Map Fusion Strategy

The strategy of depth map fusion mainly relies on geometric consistency filtering and photometric consistency filtering. Similar to the left-right disparity check in stereo tasks, MVS methods filter the scene geometries by reprojection between adjacent viewpoints. P' is obtained by mapping a pixel P in the image I_0 to the source view I_i through estimated depth $D_0(P)$, and in turn, P' can be projected back to I_0 at P'' through source depth $D_i(P')$. If the reprojected pixel location P'' and its corresponding depth $D_0(P'')$ satisfy

$$|P - P''| \leq \tau_1 \,, \tag{17}$$

$$\frac{|D_0(P'') - D_0(P)|}{D_0(P)} \leq \tau_2 \,, \tag{18}$$

we say the depth estimation of D_0 is two-view consistent with D_i. And the dynamic fusion strategy [69] can adaptively select the number of consistent viewpoints suitable for the scene scale and the total number of viewpoints.

As for photometric consistency filtering, existing methods always use the fixed parameter setting and adopt different thresholds for different scenarios. We model the full-scene depth distribution based on the Gaussian-Mixture Model in the main text, and the observation of large volume depth values tending to be concentrated in small areas is also adopted for filtering. Fig. 11 visualizes two confidence maps and the corresponding distributions. We can see that there is a large concentration of energy around the confidence close to 1 which means the network considers the depth estimates at these pixel locations to be very reliable. However, the confidence distribution of most pixels still satisfies the Gaussian distribution. Therefore, we fuse the full-scene point cloud using pixel with confidence $c \geq \mu - 3\sigma$ for each scenario. It should be noted that μ is shifted to the right and the σ is larger due to the concentration of energy at high confidence locations, but we do not modify the deviation since higher confidence is always more reliable.

III.2. More Point Clouds Reconstruction Results

Fig. 12 shows the reconstruction results of aerial photography on the BlendedMVS dataset, and Fig. 15 shows more reconstruction point clouds on the DTU dataset. Fig. 16 and Fig.17 present qualitative reconstruction point clouds on the intermediate and advanced sets of the Tanks and Temples dataset respectively. It can be seen that the proposed GeoMVSNet can produce 3D reconstruction models (point clouds) with remarkable accuracy and completeness.

III.3. Computational Effectiveness

We compare the overall score with respect to running time (Left) and GPU memory consumption (Right) in Fig.13 on the DTU dataset. Our method has a significant advantage in run-time and overall performance. This is because we do not introduce additional external dependencies, and we only need to sample a few depth hypothesis planes. Our memory consumption is comparable to existing methods, and the reason for slightly higher consumption compared to other cascade-based methods is mainly that we explicitly encode the coarse probability volumes into the cost regularization network. However, we replace the complex 3D convolutions with 2D convolutions, so there is an even lower running time.

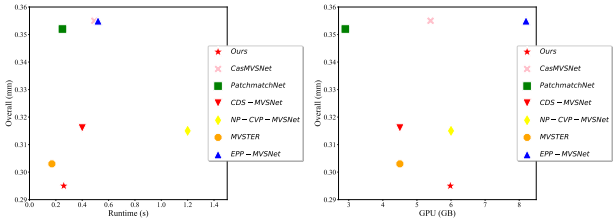

Figure 13. **Comparison of run-time and memory consumption of recent methods on the DTU dataset.**

Table 4. **Ablation results of feature fusion on the DTU dataset.**

Method	Acc. (*mm*)	Comp.(*mm*)	Overall↓ (*mm*)
a) original feat.	0.3629	0.3016	0.3323
b) branch feat.	0.3577	0.3321	0.3449
c) original + branch	0.3520	0.2893	**0.3207**

Table 5. **Ablation results of geometry embedding on the intermediate set of the Tanks and Temples dataset.**

Method	Mean↑	Family	Francis	Horse	L.H.	M60	Panther	P.G.	Train
1) w/o embedding	62.12	80.96	**65.53**	46.91	63.87	61.78	60.90	60.49	**56.53**
2) X + Y	61.56	79.99	65.41	43.69	64.63	62.27	60.05	61.48	54.92
3) Z	62.89	80.27	64.61	51.67	64.29	**63.32**	**61.55**	61.42	55.98
4) X + Y + Z	**63.52**	**81.17**	65.48	**53.46**	**65.62**	62.85	61.26	**62.15**	56.14

III.4. More Ablation Studies

The fusion of the original feature. We use the proposed two-branch geometry fusion network to integrate geometric priors contained in coarse depth maps with ordinary features extracted by the classic FPN. However, the *Branch* network itself can generate structural features for the reference image. Tab. 4 shows the comparison between a) original feature only; b) branch feature only; c) branch feature fuse with the original feature.

We can see that the branch feature itself is difficult to characterize the MVS input images well. This is due to the fact that only the feature of the reference viewpoint is extracted by the geometric prior embedded in the coarse depth map, while the corresponding source features are still extracted by the ordinary FPN. It is difficult to match the features learned by two completely unrelated neural networks. Therefore, we use the *Fusion* network to integrate the geometric structure with the original reference feature and significantly improve reconstruction completeness.

The geometry position embedding. We conduct the ablation experiments of the probability volume geometry embedding on the Tanks and Temples dataset in Tab. 5 since large-scale scenarios have rich geometric information and outdoor scenarios can better reflect the effectiveness of the method. As pointed out in the main text, the probability volume embedding strategy requires structural features as the foundation to achieve the best reconstruction quality.

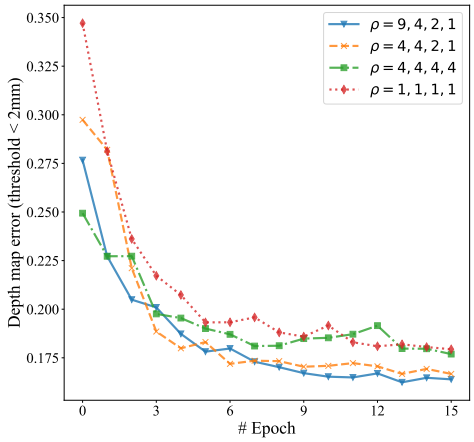

Figure 14. **Visualization of the evaluation depth map error (threshold $<$ 2mm) of the training process on the DTU dataset.**

Hence, we utilize the geometry feature fusion method in "1) w/o embedding" experiment. And we embed different positional encodings into the cost regularization network.

As shown in "2) X + Y", the pixel-wise positional encoding hardly works since the cost regularization network mainly performs matching optimization in the depth dimension and also increases the unnecessary learning burden of the network. The encoding in the depth dimension can improve the overall reconstruction quality of almost every scene as shown in "3) Z", and achieve better results with the positional enhancement of different regions as shown in "4) X + Y + Z". The probability volume geometry embedding makes full use of the rich geometric clues contained in coarse layers without introducing complex external dependencies to enhance the full-scene geometry perception.

The curriculum learning strategy. We propose to use frequency domain filtering to eliminate the high-frequency clutter textures and adopt the curriculum learning strategy to embed geometric priors into finer stages from easy to difficult. And we adjust the learning weight by modulating the cutout kernel ratio ρ.

The evaluation depth map error of the training process at different parameter settings is shown in Fig. 14. From the figure, we have the observation that the curriculum learning strategy can improve the quality of training convergence. And the frequency domain filtering strategy can effectively prevent the interference of wrong high-frequency information and obtain a more accurate depth map. However, the lack of high-frequency textures will make it difficult for the network to learn more new knowledge in later stages (green rectangle curve). Our parameter setting strategy has the most stable learning curve and the lowest depth map error, which fully embeds the geometric priors into the *Fusion* network and cost regularization network.

IV. Limitation

As aforementioned, we explicitly integrate geometric priors into the MVS network without introducing external dependencies. However, the two-branch feature fusion network and the embedding of the coarse probability volumes still increase the complexity of the cascade-based framework. How to better encode scene structures for MVS networks without introducing additional complexity is still an open problem. Besides, coarse stages only imply the geometric clues of the reference view, and how to model the geometric information of the source views is also a problem worthy of further study.

References

[66] Robert T Collins. A space-sweep approach to true multi-image matching. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 358–363. Ieee, 1996. 1, 2

[67] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3273–3282, 2019. 1

[68] Qingshan Xu and Wenbing Tao. Learning inverse depth regression for multi-view stereo with correlation cost volume. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12508–12515, 2020. 1

[69] 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 *European conference on computer vision*, pages 674–689. Springer, 2020. 3

[70] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018. 1

[71] 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 *European Conference on Computer Vision*, pages 766–782. Springer, 2020. 2

Figure 15. **Visualization on more reconstruction point clouds on the DTU dataset.**

Figure 16. **Visualization of the reconstruction point clouds on the intermediate set of the Tanks and Temples dataset.**

Figure 17. **Visualization of the reconstruction point clouds on the advanced set of the Tanks and Temples dataset.** The first and third rows are the global perspective of large-scale buildings. And the second and last rows are the zoomed-in snapshots.

Table 6. **Detailed data structures and parameters of the geometry fusion network.**

Input	Input Size	Layer	Output	Output Size
(a) $\mathcal{B}$-branch				
Conv.*				
[rgb,depth]	$4 \times H \times W$	Conv, kernel size=5, stride=1	-	$8 \times H \times W$
-	$8 \times H \times W$	BatchNorm	-	$8 \times H \times W$
-	$8 \times H \times W$	ReLU	rf	$8 \times H \times W$
ResBlock*				
rf	$8 \times H \times W$	Conv+BN+ReLU	rf1	$16 \times H/2 \times W/2$
rf1	$16 \times H/2 \times W/2$	Conv+BN+ReLU	rf2	$32 \times H/2 \times W/2$
rf2	$32 \times H/2 \times W/2$	Conv+BN+ReLU	rf3	$64 \times H/4 \times W/4$
rf3	$64 \times H/4 \times W/4$	Conv+BN+ReLU	rf4	$128 \times H/4 \times W/4$
rf4	$128 \times H/4 \times W/4$	Conv+BN+ReLU	rf5	$256 \times H/8 \times W/8$
DeResBlock*				
rf5	$256 \times H/8 \times W/8$	ConvTranspose+BN+ReLU	rfup4	$128 \times H/4 \times W/4$
rfup4+rf4	$128 \times H/4 \times W/4$	ConvTranspose+BN+ReLU	rfup3	$32 \times H/2 \times W/2$
rfup3+rf2	$32 \times H/2 \times W/2$	ConvTranspose+BN+ReLU	rfup2	$16 \times H/2 \times W/2$
rfup2+rf1	$16 \times H/2 \times W/2$	ConvTranspose+BN+ReLU	rfup1	$8 \times H \times W$
rfup1+rf	$8 \times H \times W$	ConvTranspose+BN+ReLU	$\mathcal{B}$-output	$2 \times H \times W$
Conv.*				
$\mathcal{B}$-output	$2 \times H \times W$	1st channel layer	$\mathcal{B}$-struct	$1 \times H \times W$
(b) $\hat{\mathcal{B}}$-branch				
Conv.*				
[depth,$\mathcal{B}$-struct]	$2 \times H \times W$	Conv, kernel size=5, stride=1	-	$8 \times H \times W$
-	$8 \times H \times W$	BatchNorm	-	$8 \times H \times W$
-	$8 \times H \times W$	ReLU	rf'	$8 \times H \times W$
ResBlock*				
rf'	$8 \times H \times W$	Conv+BN+ReLU	rf1'	$16 \times H/2 \times W/2$
rf1'	$16 \times H/2 \times W/2$	Conv+BN+ReLU	rf2'	$32 \times H/2 \times W/2$
[rf2,rf2']	$64 \times H/2 \times W/2$	Conv+BN+ReLU	rf3'	$64 \times H/4 \times W/4$
rf3'	$64 \times H/4 \times W/4$	Conv+BN+ReLU	rf4'	$128 \times H/4 \times W/4$
[rf4,rf4']	$256 \times H/4 \times W/4$	Conv+BN+ReLU	rf5'	$256 \times H/8 \times W/8$
DeResBlock*				
rf5+rf5'	$256 \times H/8 \times W/8$	ConvTranspose+BN+ReLU	rfup4'	$128 \times H/4 \times W/4$
rfup4'+rf4'	$128 \times H/4 \times W/4$	ConvTranspose+BN+ReLU	rfup3'	$64 \times H/4 \times W/4$
rfup3'+rf3'	$64 \times H/4 \times W/4$	ConvTranspose+BN+ReLU	rfup2' ($\hat{\mathcal{B}}$-struct)	$32 \times H/2 \times W/2$
rfup2'+rf2'	$32 \times H/2 \times W/2$	ConvTranspose+BN+ReLU	rfup1' ($\hat{\mathcal{B}}$-struct)	$16 \times H/2 \times W/2$
rfup1'+rf1'	$16 \times H/2 \times W/2$	ConvTranspose+BN+ReLU	rfup' ($\hat{\mathcal{B}}$-struct)	$8 \times H \times W$
(c) *Fusion*				
rfup2'+FPN-feat	$32 \times H/2 \times W/2$	ConvTranspose+BN+ReLU	geo-fused-feat ($\ell = 1$)	$32 \times H \times W$
rfup1'+FPN-feat	$16 \times H/2 \times W/2$	ConvTranspose+BN+ReLU	geo-fused-feat ($\ell = 2$)	$16 \times H \times W$
rfup'+FPN-feat	$8 \times H \times W$	ConvTranspose+BN+ReLU	geo-fused-feat ($\ell = 3$)	$8 \times H \times W$