

LiteMVS: Efficient Multi-View Stereo with Foundation Distillation and Expert Aggregation

Supplementary Material

1. Implementation Details

Our implementation is based on PyTorch. We adopt EfficientNetV2-S [6] as the encoder backbone, paired with a UNet++-style decoder [7]. For matching feature extraction, we employ the first two blocks of ResNet18 (R18) [3]. A comprehensive architecture description is provided in the supplementary material.

Training is performed using the AdamW optimizer [4] for 100k iterations (approximately 9 epochs) with a weight decay of 10^{-4} . We employ a step-wise learning rate schedule: 10^{-4} for the first 70k steps, 10^{-5} from 70k to 80k steps, and 10^{-6} for the remaining steps. The entire training process takes approximately 36 hours on two NVIDIA A100 GPUs (40GB). We select the model checkpoint with the lowest validation loss for final evaluation.

Input images are resized to 512×384 , and depth predictions are generated at half this resolution. During training, we apply random color jittering to brightness, contrast, saturation, and hue, with a jitter factor of $\delta = 0.2$ for all parameters. Additionally, horizontal flipping is applied with a probability of 50%.

2. Loss Function

Depth regression loss — We follow [5] and densely supervise predictions using log-depth, but use an absolute error on log depth for each scale s ,

$$\mathcal{L}_{\text{depth}} = \frac{1}{HW} \sum_{s=1}^4 \sum_{i,j} \frac{1}{s^2} |\uparrow_{gt} \log \hat{\mathbf{D}}_{i,j}^s - \log \mathbf{D}_{i,j}^{\text{gt}}|, \quad (1)$$

where we upsample each lower scale depth using nearest neighbor upsampling to the highest scale we predict at with the \uparrow_{gt} operator. We average this loss per pixel, per scale and per batch. Our experiments found this loss to perform better than the scale-invariant formulation of Eigen et al. [2], while producing much sharper depth boundaries, resulting in higher fused reconstruction quality.

Multi-scale gradient and normal losses — We follow [5] and use a multi-scale gradient loss on our highest resolution network output

$$\mathcal{L}_{\text{grad}} = \frac{1}{HW} \sum_{s=1}^4 \sum_{i,j} |\nabla \downarrow_s \hat{\mathbf{D}}_{i,j} - \nabla \downarrow_s \mathbf{D}_{i,j}^{\text{gt}}|, \quad (2)$$

where ∇ is first order spatial gradients and \downarrow_s represents downsampling to scale s . Inspired by [5] we also use a simplified normal loss, where \mathbf{N} is the normal map computed

using the depth and intrinsics (see supp. mat. for details),

$$\mathcal{L}_{\text{normals}} = \frac{1}{2HW} \sum_{i,j} 1 - \hat{\mathbf{N}}_{i,j} \cdot \mathbf{N}_{i,j}. \quad (3)$$

Multi-view depth regression loss — We use ground-truth depth maps for each source view as additional supervision by projecting predicted depth \hat{D} into each source view and averaging absolute error on log depth over all valid points,

$$\mathcal{L}_{\text{mv}} = \frac{1}{NHW} \sum_n \sum_{i,j} |\log \hat{\mathbf{D}}_{i,j}^{0 \rightarrow n} - \log \mathbf{D}_{n,i,j}^{\text{gt}}| \quad (4)$$

where $\hat{\mathbf{D}}^{0 \rightarrow n}$ is the depth predicted for the reference image of index 0, projected into source view n . This is similar in concept to the depth regression loss above, but for simplicity is applied only on the final output scale.

Total loss — Overall our total loss is:

$$\mathcal{L}_{\text{simple}} = \mathcal{L}_{\text{depth}} + \alpha_{\text{grad}} \mathcal{L}_{\text{grad}} + \alpha_{\text{normals}} \mathcal{L}_{\text{normals}} + \alpha_{\text{mv}} \mathcal{L}_{\text{mv}}, \quad (5)$$

$$\mathcal{L} = \mathcal{L}_{\text{simple}} + \lambda_{\text{rel}} \mathcal{L}_{\text{rel}} + \lambda_{\text{norm}} \mathcal{L}_{\text{norm}}, \quad (6)$$

with $\alpha_{\text{grad}} = 1.0$, $\alpha_{\text{normals}} = 1.0$, and $\alpha_{\text{mv}} = 0.2$, $\lambda_{\text{rel}} = 0.5$, $\lambda_{\text{norm}} = 0.5$, chosen experimentally using the validation set.

3. Additional Analysis of MoE Design

In this supplementary section, we provide additional ablations on the design of the proposed MoE-based cost aggregation module, including expert heterogeneity and parameter size. Unless otherwise specified, all experiments are conducted on ScanNetv2 [1] under the same training protocol as the main paper.

Table 1. **Effect of MoE Parameter Size.** Ablation results of using different parameter sizes in the proposed MoE-based cost aggregation module on ScanNetv2. The best, second-best, and third-best results are highlighted in red, orange, and yellow, respectively.

MoE Parameter Size	ScanNetv2			
	Abs Diff↓	Abs Rel↓	Sq Rel↓	$\delta < 1.25 \uparrow$
64, 64	0.0746	0.0351	0.0107	98.31
256, 256	0.0729	0.0337	0.0102	98.41
128, 128 (Ours)	0.0702	0.0311	0.0097	98.52

Effect of MoE Parameter Size. We further analyze the effect of the MoE parameter size. As shown in Table 1, increasing the parameter size from a smaller setting improves depth estimation performance, indicating that sufficient expert capacity is important for effective aggregation. However, further enlarging the MoE module does not bring additional gains and even leads to slight degradation. This suggests that the benefit of our MoE design mainly comes from effective expert specialization and adaptive aggregation, rather than simply increasing model capacity.

References

- [1] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 1
- [2] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015. 1
- [3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1
- [4] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1
- [5] Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. Simplerecon: 3d reconstruction without 3d convolutions. In *European Conference on Computer Vision*, pages 1–19. Springer, 2022. 1
- [6] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 1
- [7] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *International workshop on deep learning in medical image analysis*, pages 3–11. Springer, 2018. 1