MVBoost: Boost 3D Reconstruction with Multi-View Refinement

Supplementary Material

1. More Implementation Details

Training. In our training process, we use a fixed set of six viewpoints (front, front right, right, back, left, front left) for supervision. Our camera model employed orthographic projection. The rank of the LoRA layer in our boosted model is 32.

Metric. We evaluate both the 2D visual quality and 3D geometric quality of the generated assets. We use the same single view and then employ each model's official multi-view generation process to create their multi-view inputs, simulating real user inference scenarios. For 3D geometric evaluation, we first align the coordinate system of the generated meshes with the ground truth meshes, and then reposition and re-scale all meshes into a cube of size $[-1, 1]^3$. We report Chamfer Distance (CD) and F-Score (FS) with a threshold of 0.05, which are computed by all vertexes from the surface uniformly.

Inference Time. Since our main network structure is similar to LGM, our inference time is the same as LGM, which is within 5 seconds.

Dataset. During training, we rely on the data we generate, which consists of $100k$ samples. The evaluation is conducted using the comprehensive GSO dataset, which contains $1k$ samples.

Strength of 0.95 As noise strength increases, refinement results become closer to the original diffusion outcomes, improving quality but reducing geometric consistency. Conversely, lower noise strength leads to results more aligned with reconstruction, offering better consistency but lower quality. A noise strength of 0.95 strikes a balance between geometric consistency and image quality.

2. More Details about Method

MVBoost generates refined multi-view as pseudo-ground truth through the Multi-View Refinement Strategy. The algorithm details are presented in Algorithm 1. The symbols used in the algorithm are explained and defined in the main paper.

We leverage the generated pseudo-ground truth to boost the 3D reconstruction model. LoRA is integrated into the self-attention and cross-attention modules of the model, with its parameters trained using the refined multi-view as supervision. The algorithm details are presented in Algorithm 2. The symbols used in the algorithm are explained and defined in the main paper.

Algorithm 1: Multi-View Refinement Strategy

```
Input: Single-View Image $c$, refine strength $s$
Output: Refined Multi-View $C^{\pi}_{\uparrow} = \{c^{\pi_i}_{\uparrow}\}^n_{i=1}$
1  $C^{\pi}_T \sim \mathcal{N}(0, I)$
2  $C^{\pi} = \{c^{\pi_i}\}^n_{i=1} \leftarrow \mathcal{G}(C^{\pi}_T; c, T)$
3  $\theta \leftarrow \mathcal{R}_{\phi}(C^{\pi})$
4  $X^{\pi} = \{x^{\pi_i}\}^n_{i=1} \leftarrow \{g(\theta, \pi_i)\}^n_{i=1}$
5  $X^{\pi}_t = \{x^{\pi_i}_t\}^n_{i=1} \leftarrow \{\alpha_t x^{\pi_i} + \sigma_t \epsilon\}^n_{i=1}$
6  $t = sT$
7  $C^{\pi}_{\uparrow} = \{c^{\pi_i}_{\uparrow}\}^n_{i=1} \leftarrow \mathcal{G}(X^{\pi}_t; c, t)$
```

Algorithm 2: Boosting Reconstruction Model

```
Input: Multi-View Dataset $\mathbb{S}$, Base Model
       Parameters $\phi$.
Output: Optimized model parameters $\phi^*$.
1  Freeze the base model parameters $\phi$
2  Initialize the model trainable parameters $\phi^*$ ;
3  while not converged do
4      Sample a batch $(C^{\pi}, C^{\pi}_{\uparrow})$ from $\mathbb{S}$
5          $\theta^* \leftarrow \mathcal{R}_{\phi^*}(C^{\pi})$
6          $X^{\pi}_{\phi^*} \leftarrow g(\theta^*, \pi)$
7          $\mathcal{L} = \mathcal{L}(X^{\pi}_{\phi^*}, C^{\pi}_{\uparrow})$
8          Update $\phi^*$ with $\nabla_{\phi^*}\mathcal{L}$
9  end
10 return $\phi^*$
```

3. Experiment on Omni3D

To test the performance on real datasets, we follow the experimental setup of Instantmesh and conduct quantitative tests on the Omni3D dataset orbiting views in Table 1. The experimental results indicate that we achieved the best performance on real datasets as well.

Table 1. Quantitative results on Omni3D dataset.

Method	PSNR↑	SSIM↑	LPIPS↓	CD↓	F-Score↑
VFusion3D	17.231	0.814	0.148	0.154	0.647
LGM	17.083	0.601	0.215	0.168	0.628
InstantMesh	17.980	0.793	0.158	0.132	0.706
MVBoost (Ours)	**18.402**	**0.831**	**0.138**	**0.121**	**0.756**

4. Experiment on OpenLRM

The framework is compatible with various reconstruction models, supporting different types of 3D representations. To further illustrate the versatility of our approach, we

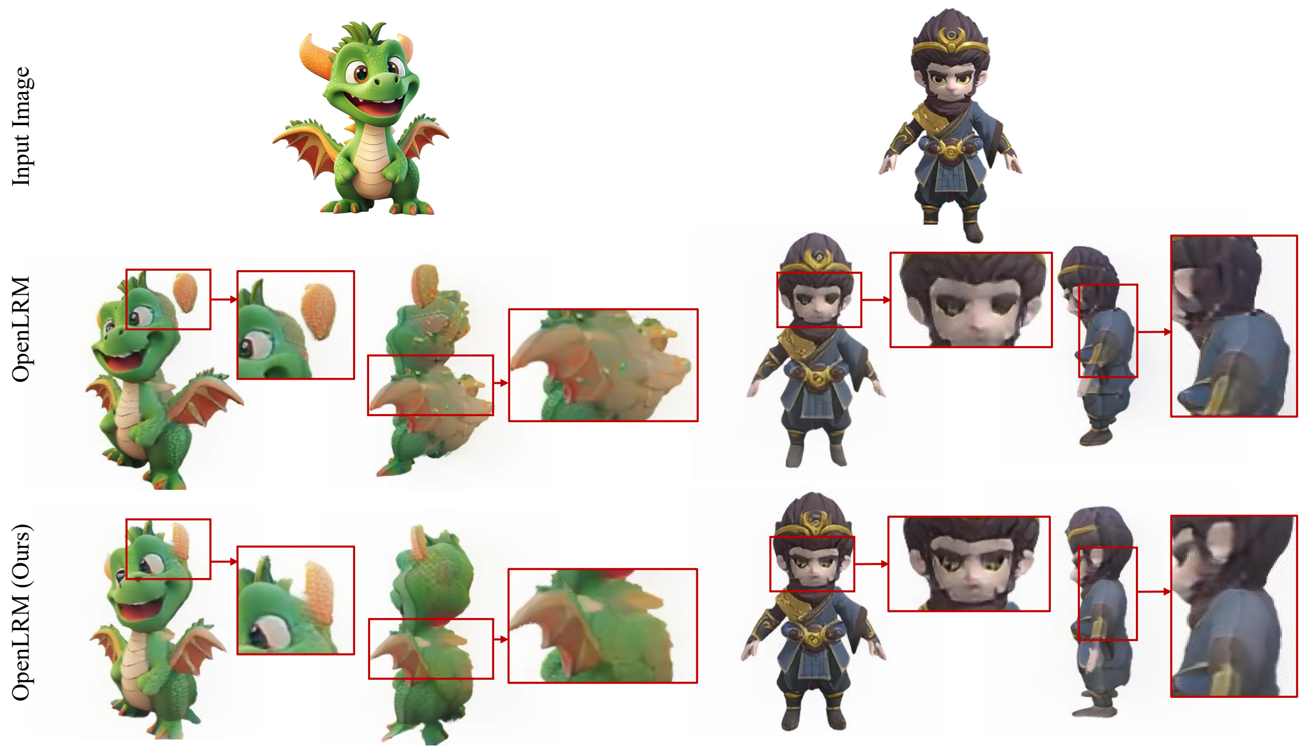

Figure 1. Qualitative experiments of boosted OpenLRM and original OpenLRM.

boost OpenLRM with multi-view refinement. We employ a dataset of $5k$ refined multi-view images, integrating LoRA layers with the rank of $r = 32$ into the self-attention and cross-attention components of the Transformer Decoder in OpenLRM. We train the boosted OpenLRM on 8 NVIDIA A100 (80G) GPUs for half a day. Qualitative results are illustrated in Figure 1, while quantitative outcomes are detailed in Table 2 and Table 3. The boost OpenLRM demonstrates superior performance over the original OpenLRM in terms of geometric and textural details.

Table 2. Visual quality comparison on Google Scanned Objects (GSO) between boosted OpenLRM and the original OpenLRM.

Method	PSNR↑	SSIM↑	LPIPS↓
OpenLRM	16.728	0.785	0.208
OpenLRM (Ours)	**17.023**	**0.832**	**0.181**

Table 3. Geometry quality comparison on Google Scanned Objects (GSO) between boosted OpenLRM and the original OpenLRM.

Method	CD↓	F-Score↑
OpenLRM	0.14786	0.6562
OpenLRM (Ours)	**0.12158**	**0.6832**

5. Limitation

We utilize a pre-trained aesthetic assessment model to evaluate and filter data based on a specific threshold. The data we generated lacks a more comprehensive basis for evaluation. In the future, we may employ a visual language model to assess our data.