DIRECT-3D: Learning Direct Text-to-3D Generation on Massive Noisy 3D Data

Supplementary Material

In this supplementary document, we provide details and extended experimental results omitted from the main paper for brevity. Specifically, Sec. 1.1 provides details of the NeRF Auto-decoder. Sec. 1.2 provides details of the 3D Super-Resolution module. Then, we cover the training details in Sec.1.3, including loss functions and warm-up training on clean data. Sec. 2 presents experiment details and hyperparameters. Sec. 3 gives additional ablation studies and more qualitative results. Finally, the limitations of our method are discussed in Sec. 4.

In addition, we provide **video results for all visualizations in the supplementary file**.

1. Model Details

1.1. NeRF Auto-decoder

We employ a NeRF Auto-decoder to extract features from the generated tri-planes and get NeRF parameters. This auto-decoder consists of several multi-layer perceptrons to process the tri-plane features $\mathbf{f}_g$ and $\mathbf{f}_c$ separately. Fig. 1 illustrates its architecture, which contains several fully connected layers with non-linear activation functions. The decoding process involves two distinct branches to handle the tri-plane features separately, ensuring that $\mathbf{f}_g$ encapsulates only the geometry information and $\mathbf{f}_c$ contains only the corresponding color features.

1.2. 3D Super Resolution

Similar to the structure of the base tri-plane diffusion model, the 3D super-resolution (SR) module also employs a U-Net model as its backbone. However, we apply only one upsampling layer that directly scales the tri-plane feature from 128^2 to 512^2. To enable efficient training with a larger batch size, we train the SR module separately. Therefore, we can directly use the saved tri-plane features during the training of the base model to train the SR module. Following cascaded image generation [6], we add Gaussian blurring and Gaussian noises to the intermediate tri-plane feature $\mathbf{f}'_{(\cdot)}$.

For training, alongside the L2 loss $\mathcal{L}_{geo}(\psi^{SR})$ and $\mathcal{L}_{col}(\phi^{SR})$ on tri-plane, we apply an entropy loss

$$\mathcal{L}_{entropy} = \rho \cdot \log_2(\rho) - (1-\rho) \cdot \log_2(1-\rho)$$

to the generated NeRF to encourage full transparent or opaque points, ensuring a smoother SR generation. Here ρ denotes the cumulative sum of density weights computed when computing NeRF parameters from tri-plane features.

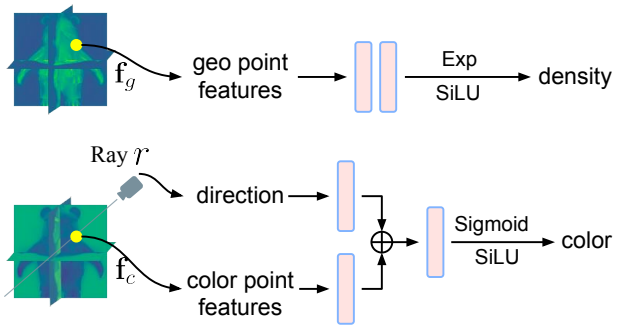

Figure 1. **Architecture of the NeRF auto-decoder.**

1.3. Training and Implementation Details

Loss function. To enable larger batch size and expedite training, we first exclude the 3D super-resolution (SR) module and train the base model end-to-end at 128^2, by minimizing the following objective:

$$\mathcal{L}_{base} = \lambda_{geo}\mathcal{L}_{geo}(\phi,\theta) + \lambda_{col}\mathcal{L}_{col}(\psi) + \lambda_{rad}\mathcal{L}_{rad}(\mathbf{f}_g,\mathbf{f}_c,\omega)$$

To speed up the convergence of tri-planes learned from multi-view images (*i.e.*, $\mathcal{L}_{rad}(\mathbf{f}_g,\mathbf{f}_c,\omega)$), we adopt prior gradient caching [4] and save the diffusion gradients $\nabla_{\mathbf{f}_g}\mathcal{L}_{geo}$ and $\nabla_{\mathbf{f}_c}\mathcal{L}_{col}$ for re-using to update the tri-plane. It enables us to update $\mathcal{L}_{rad}(\mathbf{f}_g,\mathbf{f}_c,\omega)$ multiple times in one training iteration.

Then we freeze the base tri-plane diffusion module and only train the SR module to get high-resolution generations at 512^2, with the following objective:

$$\mathcal{L}_{SR} = \lambda_{geo}\mathcal{L}_{geo}(\phi^{SR}) + \lambda_{col}\mathcal{L}_{col}(\psi^{SR}) + \lambda_{rad}\mathcal{L}_{rad}(\mathbf{f}_g,\mathbf{f}_c,\omega) + \lambda_{entropy}\mathcal{L}_{entropy}$$

In this step, we load, resize, and fine-tune the tri-plane features saved during the training of the base diffusion module. We use bilinear interpolation to scale the saved tri-planes from 128^2 to 512^2.

Warm-up training. Training the entire system is challenging due to the intricate interdependencies between different modules. Specifically, optimizing diffusion model is less effective when tri-plane $\mathbf{f}_{(\cdot)}$ in Eqn. 1 is far from convergence, but learning $\mathbf{f}_g$ with rotation θ needs a reasonably functioning diffusion model. Therefore, we warm up the model on clean and well-aligned data for the first 1/50 of the total iterations. It also defines a universal canonical pose for all objects. After that, we continue the training on all datasets with a learnable rotation parameter θ using the algorithm described in Sec. 3.2.

2. Experiment Details

2.1. Direct Text-to-3D Generation

We warm up our model on OmniObject3D [9] and a split of ShapeNet [3], which contain 6342 objects spanning 216 categories. Then we train our full model on Objaverse [5] that contains 800K+ objects.

Hyperparameters. We first train our base model for 2M iterations with a batch size of 256. Then the SR module is trained for 500K iterations with a batch size of 32. Both module are trained on 32 A100 GPUs. We set the number of channels for tri-plane features $C = 6$, and train a diffusion model with 1000 diffusion steps with linear noise schedule to generate the tri-plane features. During inference we sample 50 diffusion steps. The latent base learning rate is $1e^{-2}$ for all experiments. The learning rates for both geometry and color diffusion models are set to $1e^{-4}$, and the learning rate for NeRF auto-decoder is set to $1e^{-3}$. $\lambda_{geo} = \lambda_{col} = 5$, $\lambda_{rad} = 20$, and $\lambda_{entropy} = 0.1$. We update the tri-plane reconstructions from multi-view images 16 times per iteration for the initial 200K training iterations, and once per iteration for the subsequent training iterations. The latent base learning rate is reduced by a factor of 0.5 after 500K iterations and by a factor of 0.1 after 1M iterations.

2.2. Single-class 3D Generation

We reduce our model size to 135M parameters for a fair comparison with SSDNeRF [4] (122M). We also remove the prompt condition and train a separate model on each category following the baselines.

Hyperparameters. All models are trained for 500K iterations on 8 A100 GPUs, utilizing a batch size of 64. No SR plug-in is trained during these experiments. For cars and tables, the latent base learning rate is set to $4e^{-2}$. In the case of chairs, the latent base learning rate is set to $5e^{-3}$. The remaining hyperparameters align with those specified in direct text-to-3D generation.

2.3. Improving DreamFusion with 3D Prior

In our experiments, we sample `[animal]` from 14 animal types: *bear, corgi, dog, bird, cat, pig, elephant, horse, sheep, zebra, squirrel, chimpanzee, tiger, lion*.

Criterion for successful generation. We consider a text-to-3D generation successful when both the generated geometry and texture are consistent. Consistent geometry implies the correct number of parts is generated without missing or extra ones. Consistent texture implies the generated texture contains a consistent and plausible pattern that may appear on an actual animal of that type, regardless of the geometry.

Hyperparameters. For DreamFusion and DIRECT-3D, we run 10K iterations of optimization using the Adam optimizer [10] with a learning rate of 5×10^{-3}. Perp-Neg [1] is enabled for the 2D diffusion guidance with $w_{\text{neg}} = -4$ for all methods, which we found useful to reduce incorrect textures such as multiple head textures. We set the weight of the 3D prior SDS loss provided by DIRECT-3D to 0.01. The classifier-free guidance is set to 100 as suggested in DreamFusion [8]. We use a coarse-to-fine training process for all methods, starting from a spatial resolution of 64^2 for the first 5K iterations and increasing to 128^2 afterward. The remaining hyperparameters are set to the default values.

Figure 2. **Comparison of generated objects with and without the 3D super-resolution plug-in.** Please zoom in for better visualization.

3. Additional Experiments

3.1. Ablation on the Super-resolution Module

We employ an additional 3D super-resolution plug-in to enhance the resolution from 128^3 to 256^3. Fig. 2 compares the generated objects with and without the SR plug-in, demonstrating its effectiveness in producing high-resolution objects with reduced computational resources. However, it's worth noting that the SR plug-in may slightly alter the generated low-resolution objects and introduce additional noise.

3.2. Ablation on 3D Prior Loss Weight

We also study the impact of different 3D prior loss weights. Ablation in Fig. 3 shows that utilizing only DIRECT-3D as initialization can alleviate the Janus problem, but also results in many artifacts, while large weights could compromise the quality of the generated geometry (*e.g.*, missing rear feet in this case).

3.3. Additional Qualitative Examples

We provide additional qualitative comparisons here. Specifically, Fig. 4 provides qualitative comparison with EG3D [2] and SSDNeRF [4] on single-class 3D generation. Fig. 5 provides additional comparisons with Shap-E [7] on direct text-to-3D generation, using the same text prompts as in Shap-E. Fig. 6 provides additional qualitative results

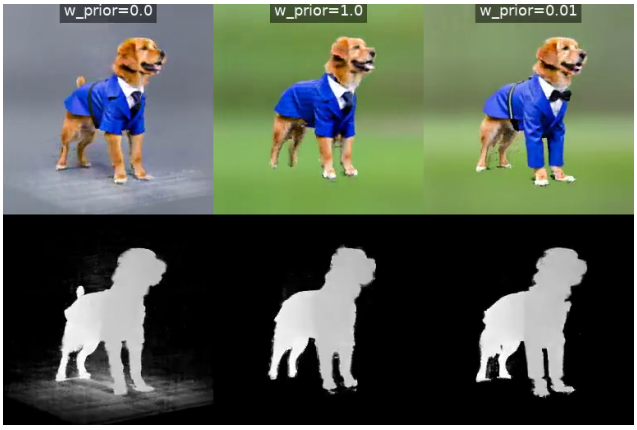

Figure 3. **Ablation of 3D prior loss weight.**

on using DIRECT-3D as a 3D prior to improve 2D-lifting text-to-3D methods such as DreamFusion [8].

4. Limitations

While DIRECT-3D consistently produces high-quality results and surpasses previous methods in single-class 3D generation and direct text-to-3D synthesis, it does exhibit certain limitations. First of all, despite the abundant geometry information provided by large-scale 3D datasets, a significant proportion of them lacks realistic textures. Additionally, the synthetic-to-real gap still persists, even for objects with nice and detailed textures. Therefore, training a 3D generative model, such as DIRECT-3D, solely on these extensive 3D datasets may result in a lack of appearance information for specific objects. One potential solution is to further fine-tune our color diffusion model on real images, which we leave for future exploration.

Secondly, the current model demonstrates limitations in compositionality. Although DIRECT-3D can generate multiple objects with close relations, such as "a house with a garden", it struggles to generate novel combinations like "an astronaut riding a horse". This issue is also observed in previous methods such as Shap-E [7]. We attribute this limitation to two main factors: (1) The scarcity of multiple objects in a single CAD model contributes to the difficulty of generating diverse objects within one tri-plane. Unlike 2D images, where multiple objects are commonly present, most 3D CAD models consist of either a single object or two or three highly related objects. (2) Current 3D datasets are still orders of magnitude smaller than their 2D counterparts, resulting in insufficient training data to effectively learn novel compositionality.

References

[1] Mohammadreza Armandpour, Huangjie Zheng, Ali Sadeghian, Amir Sadeghian, and Mingyuan Zhou. Re-imagine the negative prompt algorithm: Transform 2d diffusion into 3d, alleviate janus problem and beyond. *arXiv preprint arXiv:2304.04968*, 2023. 2

[2] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 2

[3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2

[4] Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In *International Conference on Computer Vision*, 2023. 1, 2, 4

[5] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 2

[6] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022. 1

[7] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 2, 3, 5

[8] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2022. 2, 3

[9] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 803–814, 2023. 2

[10] Xingyu Xie, Pan Zhou, Huan Li, Zhouchen Lin, and Shuicheng Yan. Adan: Adaptive nesterov momentum algorithm for faster optimizing deep models. *arXiv preprint arXiv:2208.06677*, 2022. 2

(a) EG3D

(b) SSDNeRF

(c) Ours

Figure 4. **Qualitative comparison on ShapeNet SRN Cars.** Baseline results come from the original paper of SSDNeRF [4]. Following the baseline methods, we generate and render images at 128^2.

Figure 5. **Qualitative comparison with Shap-E [7].** All text prompts are sourced from the original paper of Shap-E. For Shap-E, we use the official code and model with the default random seed. For our method, we generate objects in 128^2 without the super-resolution plug-in. All images of both methods are rendered at 256^2. Our DIRECT-3D generates 3D objects with enhanced quality in both geometry and texture.

Figure 6. **More qualitative results on using DIRECT-3D as a 3D prior for 2D-lifting methods.** Our 3D prior alleviates issues such as multiple faces and missing/extra limbs, while also improving texture quality. Please also check the video demos for a better visualization.