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

Qihao Liu¹, Yi Zhang¹, Song Bai², Adam Kortylewski³,⁴, Alan Yuille¹

¹ Johns Hopkins University  ² ByteDance  ³ Max Planck Institute for Informatics  ⁴ University of Freiburg
Motivation

- Generating diverse and high-quality 3D objects is an important task.
- Challenging due to the lack of 3D data:

<table>
<thead>
<tr>
<th>Dataset name</th>
<th>size</th>
<th>annotation</th>
<th>feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D dataset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAION</td>
<td>5B</td>
<td>Image-text pair</td>
<td>Filtered with CLIP</td>
</tr>
<tr>
<td>3D dataset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShapeNet</td>
<td>51K</td>
<td>Class name</td>
<td>Clean and aligned, can be directly used for training</td>
</tr>
<tr>
<td>Objavere-XL</td>
<td>10M</td>
<td>No annotation</td>
<td>Noisy, not aligned</td>
</tr>
</tbody>
</table>

Random samples of “dog” from the Objavere-XL dataset.
Related work

- 2D-lifting methods:
  - Pro: using 2D image diffusion models, no needs for 3D data
  - Cons:
    - Slow: optimization process
    - Janus problem (multi face problem)

- Directly train 3D generative model on clean data (proprietary data):
  - Pros:
    - Fast in 3D generation
    - More accurate 3D geometry consistency
  - Cons:
    - Single/few class generation
    - Lack of diversity
    - Hard to scale up (needs considerable efforts to collect and clean data)

DreamFusion suffers from the Janus problem

Shap-E from OpenAI is trained on extensive proprietary data: is costly to obtain, requires huge efforts to further enhance data quality
Method

- Can we directly train a 3D generative model on massive noisy and not-aligned ‘in-the-wild’ 3D data such as Objaverse-XL?

“A statue of a black dog”

Challenges:
- Training directly on non-aligned data is challenging and may result in non-convergence.
- There is no consensus on 3D data representation or network architecture that can efficiently handle high-dimensional 3D data.
- NeRF is used to represent 3D objects.
- Tri-plane features enable the use of a 2D diffusion architecture.
- An iterative optimization process in the diffusion step explicitly estimates the pose and quality of the 3D data based on the conditional density.
- We disentangle the 3D geometry and 2D color of the object, modeling them hierarchically with two separate diffusion models.
Method

Disentangled tri-plane generation:

\[ \mathcal{L}_{geo}(\phi) = \mathbb{E}_{f_g^0, \epsilon, p, t}[||\epsilon - \epsilon_\phi(f_g^t, t, \tau(p))||_2^2] \]

\[ \mathcal{L}_{col}(\psi) = \mathbb{E}_{f_c^0, \epsilon, p, f_g, t}[||\epsilon - \epsilon_\psi(f_c^t, t, \tau(p), f_g)||_2^2] \]

NeRF generation from disentangled tri-plane representation:

\[ \mathcal{L}_{rad}(f_g, f_c, \omega) = \sum_i ||\hat{y}_i - \mathcal{R}(\mathcal{D}_\omega(f_g, f_c, t_i))||_2^2 \]
Training with noisy and unaligned data:

We explicitly model the 3D rotation angle of an object as $\theta = \{\theta_{\mu}, \theta_{\sigma}\}$

$$\mathcal{L}_{geo}(\phi) = \mathbb{E}_{f_g^0, \epsilon, \rho, t}[\|\epsilon - \epsilon_{\phi}(f_g^t, t, \tau(p))\|_2^2]$$

$$\mathcal{L}_{geo}(\phi, \theta) = \mathbb{E}_{f_g^0, \theta, \epsilon, \rho, t}[\|\epsilon - \epsilon_{\phi}(f_g^t; \theta, t, \tau(p))\|_2^2]$$

We consider $\theta$ as a hidden variable and estimate it based on the conditional density.
Experiments

- Single-class 3D generation

<table>
<thead>
<tr>
<th>Method</th>
<th>Car FID (↓)</th>
<th>Car KID (↓)</th>
<th>Chair FID (↓)</th>
<th>Chair KID (↓)</th>
<th>Table FID (↓)</th>
<th>Table KID (↓)</th>
</tr>
</thead>
<tbody>
<tr>
<td>π-GAN [7]</td>
<td>36.7</td>
<td>-</td>
<td>52.71</td>
<td>13.64</td>
<td>41.67</td>
<td>13.82</td>
</tr>
<tr>
<td>EG3D [8]</td>
<td>10.46</td>
<td>4.90</td>
<td>16.54</td>
<td>8.41</td>
<td>31.18</td>
<td>11.67</td>
</tr>
<tr>
<td>DiffRF [43]</td>
<td>-</td>
<td>-</td>
<td>15.95</td>
<td>7.94</td>
<td>27.06</td>
<td>10.3</td>
</tr>
<tr>
<td>SSDNeRF [10]</td>
<td>11.08</td>
<td>3.47</td>
<td>-</td>
<td>-</td>
<td>14.27</td>
<td>4.08</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>6.90</strong></td>
<td><strong>1.84</strong></td>
<td><strong>7.01</strong></td>
<td><strong>2.12</strong></td>
<td><strong>7.26</strong></td>
<td><strong>1.89</strong></td>
</tr>
</tbody>
</table>

Table 1. Single-class generation on SRN Cars, PS Chairs, and ABO Tables. Baseline results are reported by DiffRF and SSD-NeRF. We train our model from scratch using exactly the same rendered images as the baselines. KID is multiplied by $10^3$. 

EG3D

SSDNeRF

Ours
# Experiments

- **Direct text-to-3D generation**

<table>
<thead>
<tr>
<th>Ours</th>
<th>Shap-E</th>
<th>Ours</th>
<th>Shap-E</th>
<th>Ours</th>
<th>Shap-E</th>
<th>Ours</th>
<th>Shap-E</th>
</tr>
</thead>
<tbody>
<tr>
<td>“a voxelized dog”</td>
<td></td>
<td>“an astronaut”</td>
<td></td>
<td>“a chair that looks like a tree”</td>
<td></td>
<td>“a brown boot”</td>
<td></td>
</tr>
<tr>
<td>“a boat that looks like a banana”</td>
<td></td>
<td>“a bowl of food”</td>
<td></td>
<td>“a goldfish”</td>
<td></td>
<td>“a donut with pink icing”</td>
<td></td>
</tr>
<tr>
<td>“a model of a house in Tudor style”</td>
<td></td>
<td>“a spanish galleon sailing on the open sea”</td>
<td></td>
<td>“the Statue of Liberty, aerial view”</td>
<td></td>
<td>“a yellow schoolbus”</td>
<td></td>
</tr>
<tr>
<td>“a red convertible car with the top down”</td>
<td></td>
<td>“a baby grand piano viewed from far away”</td>
<td></td>
<td>“a kingfisher bird”</td>
<td></td>
<td>“an orange road bike”</td>
<td></td>
</tr>
<tr>
<td>“a carousel with a red canopy”</td>
<td></td>
<td>“a beautiful white daisy”</td>
<td></td>
<td>“a minion”</td>
<td></td>
<td>“a lemon cut in half”</td>
<td></td>
</tr>
</tbody>
</table>

Prompts from **Shap-E**

Prompts from **DreamFusion**

**complex objects**
Experiments

- Using Ours as 3D prior to improve 2D-lifting optimization-based methods (to solve multi-face problem)
Experiments

- Using Ours as 3D prior to improve 2D-lifting optimization-based methods (to solve multi-face problem)
Experiments

- Quantitative comparison on text-to-3D

<table>
<thead>
<tr>
<th></th>
<th>More realistic</th>
<th>More detailed</th>
<th>Overall preference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shap-E [29]</td>
<td>28.4%</td>
<td>22.9%</td>
<td>26.1%</td>
</tr>
<tr>
<td>Ours</td>
<td>71.6%</td>
<td>77.1%</td>
<td>73.9%</td>
</tr>
</tbody>
</table>

Table 2. **User preference studies.** We conduct user studies on 475 prompts, including all prompts from Shap-E and 162 prompts from DreamFusion. 73.9% of users prefer ours over Shape-E.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DreamFusion-SD [47]</td>
<td>12%</td>
<td>16%</td>
<td>30%</td>
</tr>
<tr>
<td>DreamFusion-IF [47]</td>
<td>10%</td>
<td>10%</td>
<td>72%</td>
</tr>
<tr>
<td>DreamFusion-SD + Ours</td>
<td>84%</td>
<td>84%</td>
<td>98%</td>
</tr>
</tbody>
</table>

Table 3. **Improving 2D-lifting text-to-3D generation.** DIRECT-3D provides a useful 3D geometry prior, enhancing the geometry consistency and increasing the generation success rate.
Experiments

- Ablation of Automatic Alignment and Cleaning (AAC)

Figure 5. **Tri-plane feature learned with/without Automatic Alignment and Cleaning (AAC) on Objaverse.** It roughly aligns the objects to get clear tri-plane features. Unaligned objects can be captured by tri-plane representation, but the inadequate axis disentanglement makes it challenging for the diffusion model to learn.

Figure 6. **Model learned with/without AAC on Objaverse.** AAC enables direct and more efficient training on noisy, unaligned data.
Experiments

- Ablation of disentanglement

<table>
<thead>
<tr>
<th></th>
<th>Car</th>
<th>Table</th>
<th>Car + Chair + Table</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FID (↓)</td>
<td>KID (↓)</td>
<td>FID (↓)</td>
</tr>
<tr>
<td>Not Disentangled</td>
<td>9.98</td>
<td>2.96</td>
<td>12.86</td>
</tr>
<tr>
<td>Disentangled</td>
<td>6.90</td>
<td>1.84</td>
<td>7.26</td>
</tr>
</tbody>
</table>

Table 5. Improvement of Disentanglement.

![Images of a dog with different attributes: GeoDiff, GeoDiff+ColorDiff, ColorDiff, and a DSLR photo of a dog.](image)

Figure 7. Disentangling geometry and color provides a proper 3D geometrical prior, while improving the high-fidelity texture from 2D image diffusion models.
Experiments

- Ablation of prompt enrichment

Figure 8. Prompt Enrichment. FID and KID are computed on the entire test set. We provide captions with varying granularities: Coarse captions enhance object-category connections, simplifying the training, while fine-grained captions enable a better understanding of detailed features such as color and part-level information.
Experiments

- More results

“a Wall-E”

“an astronaut wearing a colorful spacesuit”

“a Transformed Bumblebee robot with intricate body details”

“an french throne chair”

“a voxelized cupcake made with LEGO”

“a biplane with yellow wings”

“a red convertible car with the top down”

“a batman mask”