Generating Images with 3D Annotations Using Diffusion Models

ICLR, 2024 (Spotlight)

Authors

Wufei Ma*, Qihao Liu*, Jiahao Wang*, Angtian Wang, Xiaoding Yuan, Yi Zhang, Zihao Xiao, Guofeng Zhang, Beijia Lu, Ruxiao Duan, Yongrui Qi, Adam Kortylewski, Yaoyao Liu*, Alan Yuille

* Equal contribution

Abstract

Diffusion models have emerged as a powerful generative method, capable of producing photos-realistic images from natural language descriptions. However, these models lack explicit control over the 3D structure of the generated image. Consequently, they struggle to achieve detailed 3D annotations for the generated images or to control interactions with specific poses and distances. In this paper, we present 3D Diffusion Style Transfer (3D-DST), which incorporates 3D geometric information into the process of generating images. In 3D-DST, we utilize 3D diffusion models by using visual prompts in addition to text prompts. We generate images of the 3D objects from their 3D shape representations (e.g., ShapeNet and ObjDataset), render them from a variety of poses and viewing distances, compute the edge maps of the rendered images, and use these edge maps as visual prompts to guide the synthesis of the objects in the generated images and obtain ground truth 3D annotations automatically. This allows us to impose a wide range of visual rules, e.g., classification and 3D pose estimation, on both the generated 3D images and the corresponding 3D shapes.

We demonstrate the effectiveness of our method on the following datasets: COCO, Shapes3D, and ObjectNet. The results show that our method significantly surpasses existing methods, e.g., 3D pose estimation on ShapeNet and COCO, in terms of accuracy. Our code is available at https://github.com/XXX/3D-DST.git.
Impressive Diffusion Models

**DALL-E 3**
“A 2D animation of a folk music band composed of anthropomorphic autumn leaves.”

**Stable Diffusion XL**
“A capybara made of lego sitting in a realistic, natural field.”
Diffusion Model Applications

Character Reference “cref”

Image Editing

“The boulevards are crowded today.”

“Photo of a cat riding on a bicycle.”

“My fluffy bunny doll.”

“a cake with decorations.”
Synthetic Data for Better Recognition

Dataset examples

ObjectNet3D
Omni3D
COCO
Synthetic Data for Image Classification

Table 1: Main Results on Zero-shot Image Recognition. All results are top-1 accuracy on test set. o: object-level. s: scene-level. f: fine-grained. t: textures. si: satellite images. r: robustness.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Task</th>
<th>CLIP-RN50</th>
<th>CLIP-RN50+SYN</th>
<th>CLIP-ViT-B/16</th>
<th>CLIP-ViT-B/16+SYN</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>o</td>
<td>70.31</td>
<td>80.06 (+9.75)</td>
<td>90.80</td>
<td>92.37 (+1.57)</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>o</td>
<td>35.35</td>
<td>45.69 (+10.34)</td>
<td>68.22</td>
<td>70.71 (+2.49)</td>
</tr>
<tr>
<td>Caltech01</td>
<td>o</td>
<td>86.09</td>
<td>87.74 (+1.65)</td>
<td>92.98</td>
<td>94.16 (+1.18)</td>
</tr>
<tr>
<td>Caltech256</td>
<td>o</td>
<td>73.36</td>
<td>75.74 (+2.38)</td>
<td>80.14</td>
<td>81.43 (+1.29)</td>
</tr>
<tr>
<td>ImageNet</td>
<td>o</td>
<td>60.33</td>
<td>60.78 (+0.45)</td>
<td>68.75</td>
<td>69.16 (+0.41)</td>
</tr>
<tr>
<td>SUN397</td>
<td>s</td>
<td>58.51</td>
<td>60.07 (+1.56)</td>
<td>62.51</td>
<td>63.79 (+1.28)</td>
</tr>
<tr>
<td>Aircraft</td>
<td>f</td>
<td>17.34</td>
<td>21.94 (+4.60)</td>
<td>24.81</td>
<td>30.78 (+5.97)</td>
</tr>
<tr>
<td>Birdsnap</td>
<td>f</td>
<td>34.33</td>
<td>38.05 (+3.72)</td>
<td>41.90</td>
<td>46.84 (+4.94)</td>
</tr>
<tr>
<td>Cars</td>
<td>f</td>
<td>55.63</td>
<td>56.93 (+1.30)</td>
<td>65.23</td>
<td>66.86 (+1.63)</td>
</tr>
<tr>
<td>CUB</td>
<td>f</td>
<td>46.69</td>
<td>56.94 (+10.25)</td>
<td>55.23</td>
<td>63.79 (+8.56)</td>
</tr>
<tr>
<td>Flower</td>
<td>f</td>
<td>66.08</td>
<td>67.05 (+0.97)</td>
<td>71.30</td>
<td>72.60 (+1.30)</td>
</tr>
<tr>
<td>Food</td>
<td>f</td>
<td>80.34</td>
<td>80.35 (+0.01)</td>
<td>88.75</td>
<td>88.83 (+0.08)</td>
</tr>
<tr>
<td>Pets</td>
<td>f</td>
<td>85.80</td>
<td>86.81 (+1.01)</td>
<td>89.10</td>
<td>90.41 (+1.31)</td>
</tr>
<tr>
<td>DTD</td>
<td>t</td>
<td>42.23</td>
<td>43.19 (+0.96)</td>
<td>44.39</td>
<td>44.92 (+0.53)</td>
</tr>
<tr>
<td>EuroSAT</td>
<td>s</td>
<td>37.51</td>
<td>55.37 (+17.86)</td>
<td>47.77</td>
<td>59.86 (+12.09)</td>
</tr>
<tr>
<td>ImageNet-Sketch</td>
<td>r</td>
<td>33.29</td>
<td>36.55 (+3.26)</td>
<td>46.20</td>
<td>48.47 (+2.27)</td>
</tr>
<tr>
<td>ImageNet-R</td>
<td>r</td>
<td>56.16</td>
<td>59.37 (+3.21)</td>
<td>74.01</td>
<td>76.41 (+2.40)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data</th>
<th>pre-trained on IN-1k?</th>
<th>0</th>
<th>1.2M</th>
<th>2.4M</th>
<th>4.0M</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None)</td>
<td></td>
<td>66.08</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>IN-1K Syn</td>
<td></td>
<td>-</td>
<td>79.00</td>
<td>80.00</td>
<td>-</td>
</tr>
<tr>
<td>IN-2K Syn</td>
<td></td>
<td>-</td>
<td>-</td>
<td>80.54</td>
<td>80.72</td>
</tr>
<tr>
<td>(None)</td>
<td>✓</td>
<td>81.30</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>IN-1K Syn</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>81.78</td>
<td>-</td>
</tr>
<tr>
<td>IN-2K Syn</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>81.87</td>
<td>81.91</td>
</tr>
</tbody>
</table>

Table 10: Results for object detection on PASCAL VOC with downstream-agnostic supervised pre-training, all results are reported in AP$_{50}$.  

Synthetic Data for Image Classification

![Graph showing top-1 accuracy (%) vs parameters (M) for different models and train set sizes.]

<table>
<thead>
<tr>
<th>Train Set (M)</th>
<th>256×256</th>
<th>1024×1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2</td>
<td>76.39 ± 0.21</td>
<td>76.39 ± 0.21</td>
</tr>
<tr>
<td>2.4</td>
<td>77.61 ± 0.08 (+1.22)</td>
<td>78.12 ± 0.05 (+1.73)</td>
</tr>
<tr>
<td>3.6</td>
<td>77.16 ± 0.04 (+0.77)</td>
<td>77.48 ± 0.04 (+0.99)</td>
</tr>
<tr>
<td>4.8</td>
<td>76.52 ± 0.04 (+0.36)</td>
<td>76.75 ± 0.07 (+0.36)</td>
</tr>
<tr>
<td>6.0</td>
<td>76.09 ± 0.08 (-0.30)</td>
<td>76.34 ± 0.13 (-0.05)</td>
</tr>
<tr>
<td>7.2</td>
<td>75.81 ± 0.08 (-0.58)</td>
<td>75.87 ± 0.09 (-0.52)</td>
</tr>
<tr>
<td>8.4</td>
<td>75.44 ± 0.06 (-0.95)</td>
<td>75.49 ± 0.07 (-0.90)</td>
</tr>
<tr>
<td>9.6</td>
<td>75.28 ± 0.10 (-1.11)</td>
<td>74.72 ± 0.20 (-1.67)</td>
</tr>
<tr>
<td>10.8</td>
<td>75.11 ± 0.12 (-1.28)</td>
<td>74.14 ± 0.13 (-2.25)</td>
</tr>
<tr>
<td>12.0</td>
<td>75.04 ± 0.05 (-1.35)</td>
<td>73.70 ± 0.09 (-2.69)</td>
</tr>
</tbody>
</table>

Synthetic Data for Segmentation

Synthetic Data with 3D Ground Truth

We are also interested in synthetic data with 3D ground truth such as 3D viewpoint, 3D location, object shape, object depth, etc.
Synthetic Data with 3D Ground Truth

We are also interested in synthetic data with 3D ground truth such as 3D viewpoint, 3D location, object shape, object depth, etc.
Synthetic Data with 3D Ground Truth
Standard Diffusion Model

Denoising diffusion models consist of two processes:

- Forward diffusion process that gradually adds noise to input
- Reverse denoising process that learns to generate data by denoising

Credit: CVPR 2022 Tutorial on diffusion models: [link](#)
Standard Diffusion Model

During generation, we gradually denoise the latent with an iterative process.

\[ z_{t-1} = \epsilon(z_t, t), \quad t = T, \ldots, 1 \]

Here \( \epsilon \) is often modeled with a trainable network with U-Net architecture.
During generation, we gradually denoise the latent with an iterative process.

\[ z_{t-1} = \epsilon(z_t, t), \quad t = T, \ldots, 1 \]

Here \( \epsilon \) is often modeled with a trainable network with U-Net architecture.

To generate images with desired contents, we add **text-conditioning** with cross-attention layers.

\[ z_{t-1} = \epsilon(z_t, \mathcal{T}, t), \quad t = T, \ldots, 1 \]
Text Conditioning

To generate images with desired contents, we add **text-conditioning** with cross-attention layers.

\[ z_{t-1} = \epsilon(z_t, T, t), \quad t = T, \ldots, 1 \]
Visual Conditioning

Furthermore, ControlNet adds visual conditioning to text-to-image diffusion models by adding visual features via **zero convolutions** (layers initialized with zero weights).

The advantage is that the base text-to-image diffusion model is fixed, and we achieve various visual conditioning by training only an adapter on top of the large pretrained diffusion model.

ControlNet

ControlNet

LooseControl

(C2) 3D Box Control

Figure credit: Olaf Dünkel.
3D Words

Synthetic Data with 3D Ground Truth
Diverse prompts improve the realism and diversity of the synthetic images. Models trained on such images are found to be more robust.

**3D-DST – Our Method**

1. Keywords from ShapeNet / Objaverse
2. Class name
3. LLM generation
Our 3D-DST
Removing Biases in Object Viewpoints

Viewpoint distribution of cars and buses from synthetic images generated by a text-to-image diffusion model and our 3D-DST.
Analyzing 3D Consistencies
Analyzing 3D Consistencies

Human evaluation on the consistencies of 3D viewpoints show that about 75% of the images produced by our 3D-DST model have correct 3D annotations for downstream training.
Failure Modes

Failure models such as guitars (top) from side view and taxi cabs (bottom) from bottom view.
K-Fold Consistency Filter (KCF)
K-Fold Consistency Filter (KCF)

With KCF we can increase the success rate of our 3D-DST model by around 6%.

- School bus: 70% to 80%
- Guitar: 66% to 76%
- Taxi: 86% to 89%
K-Fold Consistency Filter (KCF)

The pose estimators are not robust.
Main Results

- **Classification on ImageNet-100 and ImageNet-R.**
- 3D pose estimation on PASCAL3D+ and OOD-CV.
- 3D object detection on Omni3D.
- Ablation study on image generation.

### Table 1: Image classification accuracy (%) on ImageNet-100 (ID) and ImageNet-R (OOD) using representative network architectures, ResNet and ViT. We compare the performances when models are (1) trained purely on the target dataset, (2) pre-trained on Text2Img (He et al., 2023) data, which does not have 3D control, and then finetuned on the target dataset, (3) pre-trained on 3D-DST data, and finetuned on the target dataset. Experiments show that our 3D-DST data can help boost the classification accuracy of both models on both ID and OOD cases by a large margin.
Main Results

- Classification on ImageNet-100 and ImageNet-R.
- **3D pose estimation on PASCAL3D+ and OOD-CV.**
- 3D object detection on Omni3D.
- Ablation study on image generation.

<table>
<thead>
<tr>
<th>Methods</th>
<th>In-distribution (ID)</th>
<th>Out-of-distribution (OOD)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc@π/6</td>
<td>Acc@π/18</td>
</tr>
<tr>
<td>ResNet</td>
<td>82.33</td>
<td>52.60</td>
</tr>
<tr>
<td>ResNet w/ AugMix (2020)</td>
<td>82.72 (79.39)</td>
<td>53.89 (71.39)</td>
</tr>
<tr>
<td>ResNet w/ 3D-DST (ours)</td>
<td>84.22 (71.89)</td>
<td>56.52 (73.92)</td>
</tr>
<tr>
<td>NeMo (2021)</td>
<td>82.23</td>
<td>57.12</td>
</tr>
<tr>
<td>NeMo w/ AugMix (2020)</td>
<td>83.11 (79.88)</td>
<td>58.22 (71.19)</td>
</tr>
<tr>
<td>NeMo w/ 3D-DST (ours)</td>
<td>85.70 (72.47)</td>
<td>62.51 (75.49)</td>
</tr>
</tbody>
</table>

Table 4: Robust 3D pose estimation on ID (PASCAL3D+ & ObjectNet3D (Xiang et al., 2016; 2014)) and OOD (OOD-CV (Zhao et al., 2022)). We experiment with a classification-based pose estimation method, ResNet, and a 3D compositional model, NeMo (Wang et al., 2021). Experimental results demonstrate that our DST synthetic data can effectively improve 3D pose estimation performance on both ID and OOD benchmarks.
Main Results

• Classification on ImageNet-100 and ImageNet-R.

• 3D pose estimation on PASCAL3D+ and OOD-CV.

• 3D object detection on Omni3D.

• Ablation study on image generation.

### Results

<table>
<thead>
<tr>
<th>Methods</th>
<th>AP2D</th>
<th>AP3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>CubeRCNN (Brazil et al., 2023)</td>
<td>41.50</td>
<td>41.65</td>
</tr>
<tr>
<td>w/ DST-3D (ours)</td>
<td>42.34</td>
<td>42.74</td>
</tr>
<tr>
<td>w/ DST-3D + camera aug (ours)</td>
<td><strong>42.86</strong></td>
<td><strong>43.19</strong></td>
</tr>
</tbody>
</table>
Main Results

- Classification on ImageNet-100 and ImageNet-R.
- 3D pose estimation on PASCAL3D+ and OOD-CV.
- 3D object detection on Omni3D.
- **Ablation study on image generation.**
Code & Data Release

• **Code:**
  - Synthetic data rendering and generation
  - Prompt completion with LLM
  - K-fold consistency filtering (KCF)

• **Dataset:**
  - DST data for image classification
  - DST data for pose estimation
  - Aligned 3D models for each category
Future Work

3D-DST for animals.

1. 3D consistency?
2. SMAL consistency?
3. Background and foreground diversity

Figure credit: Jiawei Peng.
Future Work

3D-DST for OOD robustness evaluation.

1. Evaluating OOD robustness to snow, rain, fog, etc.

2. Continuous “sliders”

Future Work

3D-DST for multi-category multi-object scenes.

1. Broader applications
2. 3D consistency
3. Temporal consistency

Thanks