WarpingGAN: Warping Multiple Uniform Priors for Adversarial 3D Point Cloud Generation

Yingzhi Tang[1*], Yue Qian[1*], Qijian Zhang[1], Yiming Zeng[1], Junhui Hou[1†], Xuefei Zhe[2]

[1]City University of Hong Kong [2]Tencent AI lab ([*] means equal contribution, † indicates corresponding author)

CVPR JUNE 19-24 2022 NEW ORLEANS LOUISIANA

Problem Definition and Contribution

Definition: Given a latent code $\mathbf{z} \in \mathbb{R}^C$ following a Gaussian distribution, the generator denoted as $\mathcal{F}(\cdot)$ attempts to produce a point cloud $\mathbf{P} \in \mathbb{R}^{N\times 3}$ which shares the same statistical distribution as a real dataset $\{\widetilde{\mathbf{P}}\}$.

Key Contributions:

- we investigate the GAN-based 3D point cloud generation from the new perspective of *unified local-warping*, leading to WarpingGAN featured with lightweight, high efficiency, and flexible output; and
- by taking advantage of the inherent design of the discriminator without introducing additional complex operations, we propose a stitching loss tailored to WarpingGAN to boost the generator; and
- we conduct extensive experiments and analysis to demonstrate the superiority of WarpingGAN over state-of-the-art methods.

Problem Formulation

Main idea: Motivated by the folding-style design, we proposed a warping-based generator. The generator has two modules, i.e., code enhancement and unified local-warping. The code enhancement module (composed of five fully-connected layers) transforms $\mathbf{z}$ to $\widetilde{\mathbf{z}} \in \mathbb{R}^D$ of a higher dimension ($D > C$), to make the code have enough implicit semantic information to represent shapes faithfully. The unified local-warping module first partitions $\widetilde{\mathbf{z}}$ into M local codes of an equal length denoted as $\{\widetilde{\mathbf{z}}^j \in \mathbb{R}^{D/M}\}_{j=1}^M$ and then concatenate each of the local codes with the global shape information, leading to $\mathbf{z}^j = [\widetilde{\mathbf{z}}^j \ \widetilde{\mathbf{z}}] \in \mathbb{R}^{D(M+1)/M}$. Finally, we concatenate $\mathbf{z}^j$ with each uniform prior $\mathbf{u}_i^j$ and adopt unified local-warping twice.

Method

Below is the flowchart WarpingGAN, which consists of a warping-based generator and a discriminator. The generator takes a latent code and M uniform priors as input, and aims to generate a 3D point cloud shape. The discriminator adopts the PointNet as the backbone, which takes the generated and real 3D point cloud shapes as input to perform binary classification. Owing to the max-pooling operation, we retrieve a small number of *critical points* from the input point cloud for computing the stitching loss, which is used to eliminate the gaps between different priors. Stitching loss is formulated as:

$$\mathcal{L}_s = \left(\frac{1}{Q}\sum_{i=1}^{Q}\mathrm{var}(\mathbf{q}_i,\mathcal{N}(\mathbf{q}_i)) - \frac{1}{\widetilde{Q}}\sum_{i=1}^{\widetilde{Q}}\mathrm{var}(\widetilde{\mathbf{q}}_i,\mathcal{N}(\widetilde{\mathbf{q}}_i))\right)^2$$

where $\mathrm{var}(\mathbf{q}_i,\mathcal{N}(\mathbf{q}_i)) = \sum_{k=1}^{K}\frac{(d_i^k-\bar{d}_i)^2}{K}$ and $\bar{d}_i$ is the mean value of $\{d_i^k\}_{i=1}^{Q}$. We adopt WGAN loss and 1-Lipschitz constraint to train WarpingGAN.

Loss function for the generator:

$$\mathcal{L}_g = -\mathbb{E}_{\mathbf{P}\sim\mathbb{P}_\mathbf{P}}[\mathcal{D}(\mathbf{P})] + \lambda_s\mathcal{L}_s$$

Loss function for the discriminator:

$$\mathcal{L}_d = \mathbb{E}_{\mathbf{P}\sim\mathbb{P}_\mathbf{P}}[\mathcal{D}(\mathbf{P})] - \mathbb{E}_{\mathbf{P}\sim\mathbb{P}_{\widetilde{\mathbf{P}}}}[\mathcal{D}(\widetilde{\mathbf{P}})] + \lambda_{gp}\mathbb{E}_{\widehat{\mathbf{P}}\sim\mathbb{P}_{\widehat{\mathbf{P}}}}[(\|\nabla_{\widehat{\mathbf{P}}}\mathcal{D}(\widehat{\mathbf{P}})\|_2 - 1)^2]$$

Experiments & Results

Visual Results:

Validation of code enhancement and global shape code:

wo/w code enhancement — *wo/w* global shape code

Ablation studies:

2D vs. 3D priors — non-uni vs. uni priors

4 priors — 16 priors — 64 priors

Folding-based vs. WarpingGAN — Atlas-based vs. WarpingGAN

The effectiveness of the stitching loss:

Flexibility illustration:

1024 points — 2048 points — 4096 points

Video Demo:

Github Webpage: