2023 IEEE/CVF International Conference on Computer Vision (ICCV) | 979-8-3503-0718-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCV51070.2023.01311

DiffFacto: Controllable Part-Based 3D Point Cloud Generation with Cross Diffusion

George Kiyohiro Nakayama[1] Mikaela Angelina Uy[1] Jiahui Huang[2]
Shi-Min Hu[2] Ke Li[3] Leonidas Guibas[1]
[1]Stanford University [2]Tsinghua University [3]Simon Fraser University

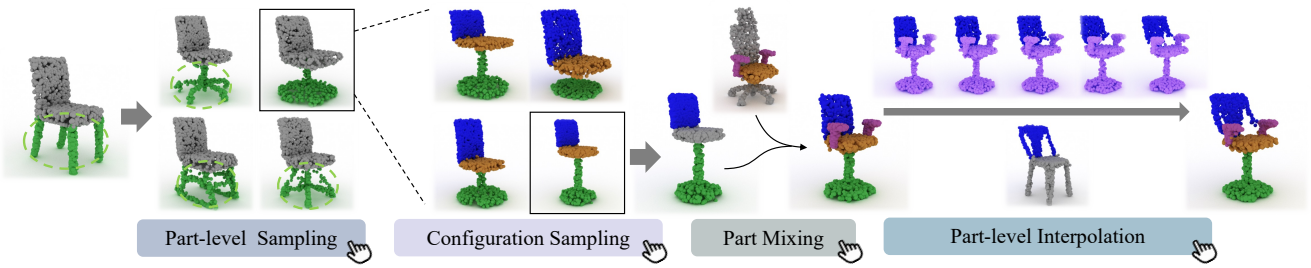

Figure 1: **DiffFacto**: Our approach tackles the task of controllable part-based point cloud generation, where we are able to generate novel shapes - novel configurations of novel parts. Our probabilistic generative model learns a *factorized* representation of shapes with a *cross diffusion* network allowing for control. We demonstrate that **DiffFacto** not only enables controllable generation but also various shape editing tasks.

Abstract

While the community of 3D point cloud generation has witnessed a big growth in recent years, there still lacks an effective way to enable intuitive user control in the generation process, hence limiting the general utility of such methods. Since an intuitive way of decomposing a shape is through its parts, we propose to tackle the task of controllable part-based point cloud generation. We introduce ***DiffFacto****, a novel probabilistic generative model that learns the distribution of shapes with part-level control. We propose a* factorization *that models independent part style and part configuration distributions, and present a novel* cross diffusion *network that enables us to generate coherent and plausible shapes under our proposed factorization. Experiments show that our method is able to generate novel shapes with multiple axes of control. It achieves state-of-the-art part-level generation quality and generates plausible and coherent shape while enabling various downstream editing applications such as shape interpolation, mixing, and transformation editing. Please visit our project webpage at* https://difffacto.github.io/

1. Introduction

3D shape generation [45] is an important and popular task , where point clouds are one popular representation [46, 5, 52, 28] – due to their *simple yet powerful* expressivity as well as data availability, *i.e.* just a set of points and can directly be acquired by sensors. However, the generation of arbitrary plausible shapes is often of limited utility, as users often have a conceptual design idea and of what they want to generate.

A prerequisite of shape generation is to be able to learn a space of all possible shapes. To represent this space, one parsimonious way is to represent them as a combination of simpler atoms, known as *parts*. In this flavor, we propose the task of *controllable part-based generation*, which aims to generate plausible novel shapes with user control over individual parts. As mentioned before, a shape is a combination of parts, thus a 'novel shape' can be defined in three different ways: (1) *novel* configurations of existing parts, (2) existing configurations of *novel* parts, and (3) *novel* configurations of *novel* parts.

The first is explored in existing graphics literature such as part retrieval [41] and shape assembly [50], while the second can be tackled by existing generation methods [46, 5, 52] trained on parts. In contrast, we tackle the third, which is the more challenging case subsuming the first and second. A challenge arises because a shape is a combination of novel parts, leading to an exponential explosion of plausible shapes while having only limited training data. A further challenge stems from enabling control as this requires an approach that can vary individual parts and configurations while still generating plausible shapes.

To this end, we introduce a new method that tackles this task in a principled way by building a *probabilistic generative model* that learns the distribution of shapes while enabling control on parts and configurations. Specifically,

2380-7504/23/$31.00 ©2023 IEEE
DOI 10.1109/ICCV51070.2023.01311

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:30:43 UTC from IEEE Xplore. Restrictions apply.

we propose a *factorization* that decomposes the shape space into (i) the individual canonicalized (semantic) parts, and (ii) their transformations (position and size). These factors can be sampled or encoded independently, allowing for different modes of control in generation and intuitive editing.

Our approach learns independent latent spaces for each canonicalized (semantic) part through *part stylizers*. Then conditioned on the canonicalized parts, we also introduce learn a *transformation sampler* that learns a distribution of part configurations. Naive approaches can result in mode collapse since multiple parameter configurations can output a valid shape, if conditioned only on canonicalized parts. We leverage on a sampling-based approach to learn a multi-modal distribution of part configurations through conditional Implicit Maximum Likelihood [24] (cIMLE).

To generate plausible shapes through independently sampled factors, we also introduce our *cross diffusion* network that allows for the learning of a better shape distribution under our proposed factorization. Our cross-attention diffusion network, conditions on the proposed factors, *i.e*. independent part style and transformations, in the reverse diffusion process. Our design allows each generated point in the point cloud to be informed of both the global shape as well as the local part, resulting in more plausible and coherent output shapes while still enabling control. Moreover, we also introduce a generalized forward diffusion kernel that allows the explicit encoding of each part transformations, enabling better shape reconstruction and transformation extrapolation.

We dub our method **DiffFacto**, for **facto**rized represention with cross **diff**usion. To our best knowledge, we are the first to introduce a factorized representation that allows for control in both part styles and part configurations as we model independent part style distributions and transformation distribution, enabling each to be independently sampled. Experiments show that our approach achieves better intra-part and inter-part level scores compared to baselines. We also show that our approach generates novel and coherent shapes through a segmentation-based plausibility experiment and human study. Furthermore, we demonstrate that our approach also allows for controllable and localized shape editing on various applications such as part-level shape interpolation, shape mixing and transformation editing.

2. Related Work

Point Cloud Generative Models. The literature mainly targets at an accurate modeling of the underlying data distribution, using probabilistic tools and parameterized deep networks developed in variational auto-encoders (VAEs) [19] used by PSG [10], generative adversarial networks (GANs) [12] used by PointGAN [1], auto-regressive models [4] used by PointGrow [39], normalizing flows [35] used by PointFlow [46] or Softflow [18], *etc*., with proper conditioning [38, 30, 24]. Notably, the most recent success of Diffusion Models [16] for image synthesis [36] is further expanded to the domain of 3D point clouds generation [28, 53, 5, 52, 33, 17]. Among this line of works, PVD [54] and LION [52] treat the full point cloud as a *single* sample from the learned distribution in either primal or latent space. However, the points themselves from a cloud can be viewed as samples from a geometric distribution, which is explicitly modeled by, *e.g*., DPM [28] or ShapeGF [5], and PointFlow [46] as *distribution of distributions* [46]. This enables the desirable capability to generate *arbitrary* number of points from a single shape without the need of re-training. Our work takes this approach and additionally builds a *factorized* prior to sample from allowing for part-level control, which in contrast to all previous works that only model a *single* latent space and can only generate a full shape.

Structure-aware Shape Generation. Different from *full-shape* generation, structure-aware methods enrich the synthesized geometry with useful structural information and allow for easy user manipulations [7]. Related literature has explored different structure representations, such as segmented parts [32, 21, 26], relationship graphs [11, 44], or layers/hierarchies [22, 13, 47], generated using either auto-regressive models [43, 33], recursive networks [22], or hierarchical models [47, 42]. Among these works, SPAGHETTI [14] uses cross-attention to mix the part latents that are decoded into an implicit shape representation. SP-GAN [25] employs a spherical proxy geometry for even finer-grained controls. Nevertheless, most of the above existing methods model the shape prior as a single latent code, and user controls are *detached* from the generative process. They hence rely on post-optimization/inversion [13] steps to maintain the shape plausibility. In contrast, our *factorized* prior allows for explicit part-based control and sampling *during* the generative process.

Shape Editing. In graphics, *low-level* shape editing usually involves computing and modifying geometric handles such as cages [37, 9] or skeletons [51, 3]. On the other hand, the exploitation of *high-level* semantics allow for more intuitive controls of the shape such as mixing and stitching semantic parts from a shape collection [50], tweaking the scales and deformations of the bounding boxes [27, 40], changing the status of a coarse proxy geometry [48, 29], or using natural languages [20]. They typically however start from an *existing source shape* which is edited to a desired target. In contrast, in addition to editing existing geometry, our method can also create *novel* shapes in a *controllable* manner with our introduced factorization, and enabling a wider range of applications than the prior arts.

3. Problem Overview

Our goal is to learn a *controllable* generative model on 3D shapes. Given a set of shapes $\mathbf{S} = \{S^{(i)}\}$, we want to learn a distribution $P(\mathbf{S})$ from which we can sample with part-level control. To generate realistic and plausible shapes, a shape *prior* is commonly learned. Existing works [46,

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:30:43 UTC from IEEE Xplore. Restrictions apply.

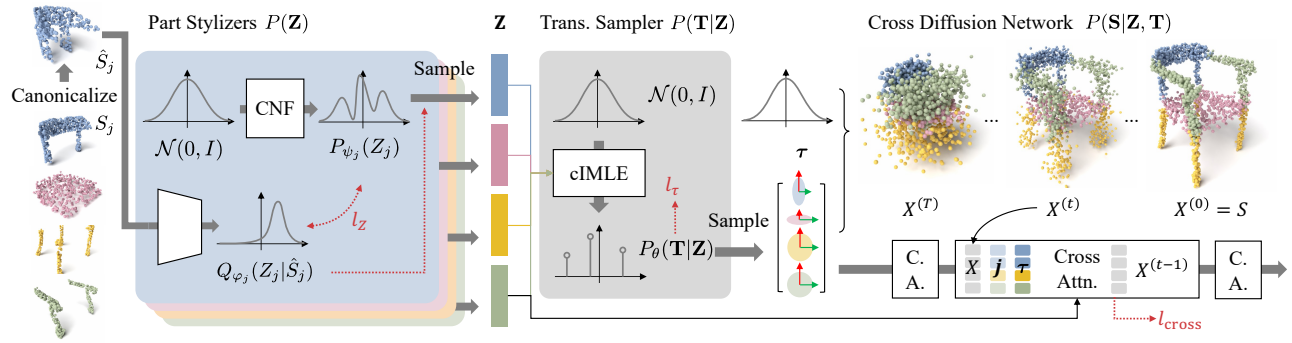

Figure 2: **Method Overview.** We factorize the 3D shape distribution into three key components, containing m **part stylizers** for each part that model the shape prior $P(\mathbf{Z})$, a **transforamtion sampler** that models the conditional distribution of transformations given the part latents $P(\mathbf{T}|\mathbf{Z})$, and a **cross diffusion network** that samples the point cloud jointly considering the part geometry and their configurations $P(\mathbf{S}|\mathbf{Z}, \mathbf{T})$. Red dashed lines indicate losses incorporated in the training stage. For definitions of variables, see Sec. 3.

5, 28, 15] model the shape prior as one random variable $w \sim P(W)$, *i.e.* a single global latent code, which *does not* allow for any level of structure-aware control on the shape being sampled/generated.

We introduce a **factorized** representation of the prior in order to obtain more localized control of the generation process. An intuitive granularity for users to control shape generation is through a natural decomposition of shapes into semantic parts [31], which we leverage on in our proposed factorization. For the rest of the paper, we assume that shapes in $\mathbf{S}$ from a given category have a predefined set of m semantic parts.

A shape $S^{(i)}$ is decomposed into its semantic parts $\{S_j^{(i)}\}_{j=1,...,m}$, where $S_j^{(i)}$ is the geometry of each shape part the superscript is omitted in what follows for brevity. We further factorize S_j into its *canonicalized geometry*, $\hat{S}_j$, *i.e.* **part style**, and its corresponding **instancing transformation**, $T_j \in \mathbb{R}^{D_1}$. We then model a independent distributions $P(Z_j)$ for each canonicalized part, where $Z_j \in \mathbb{R}^{D_2}$ $j = 1, ..., m$ is the canonicalized part latents, which we call as part stylizers. For simplicity, we use bold type for random variables and samples on the collection for all parts, e.g. $\mathbf{Z} = \{Z_j\}$, uppercase letters are random variables and lowercase letters are their corresponding samples, e.g. $z_j \sim Z_j, \tau_j \sim T_j$. Conditioned on all the part style latents $\mathbf{Z}$, we further model a distribution on their transformations, $P(\mathbf{T}|\mathbf{Z})$, where $\mathbf{T} = \{T_j\}$. The shape prior distribution is then the *joint* distribution of individual part styles and their transformations. Our proposed shape factorization is then given as:

$$P(\mathbf{S}) = \int\int P(\mathbf{S}|\boldsymbol{z}, \boldsymbol{\tau}) P(\boldsymbol{\tau}|\boldsymbol{z}) P(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z} \, \mathrm{d}\boldsymbol{\tau} \,, \quad (1)$$

where our shape prior is now *factorized* as $P(\mathbf{Z}, \mathbf{T}) = \prod_{j=1}^{m} P(Z_j) P(\mathbf{T}|\mathbf{Z})$. Our factorization allows for control in our learned generative model, since we are able to independently sample from each part style latent distribution $z_j \sim P(Z_j)$ and a valid set of transformations $\boldsymbol{\tau} \sim P(\mathbf{T}|\mathbf{Z})$, thus introducing multiple different knobs for variation. We detail our joint prior $P(\mathbf{Z}, \mathbf{T})$ in § 4.2. To capture and generate more plausible and holistic shapes, we further introduce our *cross diffusion network* that models $P(\mathbf{S}|\mathbf{Z}, \mathbf{T})$, a conditional distribution of points on the shape's surface. We elaborate our diffusion-based network and also introduce a generalized forward kernel in § 4.3. We showcase that our proposed factorization not only allows us to sample and generate diverse and plausible shapes, but also enables multiple shape editing and variation applications in § 4.4.

4. Method

We learn a shape distribution $P(\mathbf{S})$ given segmented shapes, represented as a point cloud $S = \{x_k\}_{k=1,...,N}$, $x_k \in \mathbb{R}^3$ with their semantic parts $\{S_j\}_{j=1,...,m}$, where $S = \cup_{j=1}^{m} S_j$. Each part S_j is decomposed into its part style $\hat{S}_j$ and transformation τ_j. Concretely, the part style $\hat{S}_j$ is the canonicalized part geometry given by the transformation $\tau_j \in \mathbb{R}^6$ representing the shift $(c_j \in \mathbb{R}^3)$ and axis-aligned scale $(s_j \in \mathbb{R}^3_{>0})$, giving us:

$$S_j = \mathrm{Diag}(s_j)\hat{S}_j + c_j.$$

Our method models the data distribution $P(\mathbf{S})$ with a factorized joint prior $P(\mathbf{Z}, \mathbf{T})$ and has three components: (i) independent **part stylizers** that model $P(Z_j)$ for $j = 1, ..., m$, representing part styles, (ii) a **transformation sampler** that models a distribution $P(\mathbf{T}|\mathbf{Z})$ of the part transformations conditioned on their styles, and (iii) a **cross diffusion network** that models the conditional distribution $P(S|\mathbf{Z}, \mathbf{T})$ of points on the surface of the shape given the factorized prior. Fig. 2 illustrates our method overview.

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:30:43 UTC from IEEE Xplore. Restrictions apply.

4.1. Training Objective

We first derive the training objective for our factorized joint prior $P(\mathbf{Z}, \mathbf{T})$. We maximize the likelihood of our learned distribution $P(\mathbf{S})$ through the evidence lower bound. We let $\boldsymbol{\psi}, \theta, \phi$ be the model parameters of our part stylizer, transformation sampler and cross-diffusion network, respectively. In addition, we assume an evidence distribution $Q_{\boldsymbol{\varphi}}(\mathbf{Z}, \mathbf{T}|S)$ that can be formulated as:

$$\begin{aligned} Q_{\boldsymbol{\varphi}}(\mathbf{Z}, \mathbf{T}|S) &= Q_{\boldsymbol{\varphi}}(\mathbf{Z}|S)\, Q(\mathbf{T}|S) \\ &= Q(\mathbf{T}|S) \prod_{j=1}^{m} Q_{\varphi_j}(Z_j|S_j) \\ &= \prod_{j=1}^{m} Q_{\varphi_j}\left(Z_j|\hat{S}_j\right), \end{aligned} \tag{2}$$

where $Q(\mathbf{T}|S)$ is deterministic since we assumed known segmentation $S = \{S_j\}$, and we assume $Z_{j_1} \perp Z_{j_2}$ $\forall j_1, j_2 \in \{1, \ldots, m\}$, $j_1 \neq j_2$, *i.e.* $\mathbf{Z}$ and $\mathbf{T}$ are conditionally independent given S. Thus, the evidence lower bound (ELBO) for the likelihood can be derived as:

$$\begin{aligned} &\mathbb{E}_S\left[\log P_{\phi,\boldsymbol{\psi},\theta}(S)\right] \\ &= \mathbb{E}_S\left[\log \iint P_\phi(S|\boldsymbol{z},\boldsymbol{\tau})\, P_{\boldsymbol{\psi},\theta}(\boldsymbol{z},\boldsymbol{\tau})\, \mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{\tau}\right] \\ &= \mathbb{E}_S\left[\log \iint \frac{P_\phi(S|\boldsymbol{z},\boldsymbol{\tau})\, P_{\boldsymbol{\psi}}(\boldsymbol{z}) P_\theta(\boldsymbol{\tau}|\boldsymbol{z})}{Q_{\boldsymbol{\varphi}}(\boldsymbol{z},\boldsymbol{\tau}|S)} Q_{\boldsymbol{\varphi}}(\boldsymbol{z},\boldsymbol{\tau}|S)\mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{\tau}\right] \\ &\geq \mathbb{E}_{S,\boldsymbol{z}}\Big[\underbrace{\log P_\phi(S|\boldsymbol{z},\boldsymbol{\tau})}_{\mathcal{L}_{\text{recon}}} + \underbrace{\sum_{j=1}^{m} \log \frac{P_{\psi_j}(z_j)}{Q_{\varphi_j}(z_j|\hat{S}_j)}}_{\mathcal{L}_{\mathbf{Z}}} + \underbrace{\log P_\theta(\boldsymbol{\tau}|\boldsymbol{z})}_{\mathcal{L}_{\boldsymbol{\tau}}}\Big], \end{aligned} \tag{3}$$

where the inequality is through Jensen's (see supplement). We now go through each of our components, and how they are trained to maximize this derived training objective with the corresponding loss functions (denoted as $l_{(\cdot)}$).

4.2. Factorized Joint Prior $P(\mathbf{Z}, \mathbf{T})$

Our factorization allows us to define multiple random variables to control the generative model and output plausible shapes. These random variables can be independently sampled and are the 'control knobs' in the generation process, thus allowing for user-controllability. These random variables are Z_j to control per-part style through our part stylizers and $\mathbf{T}$ that controls the set of part transformation through our transformation sampler. Moreover, this also allows the network to consolidate part information, *e.g.* the canonicalized geometry of a rectangular back of the bench is the same as a dining chair when transformation is factored out. This decomposition also enables resampling certain part geometries Z_j while keeping the rest fixed, creating variations in part configurations by resampling $\mathbf{T}$, and local shape editing through encoding and modifying z_j or τ_j.

Part Stylizer. The part stylizer learns to model the independent distributions $P(\hat{S}_j)$ of part styles for $j = 1, ..., m$. Each $\hat{S}_j$ is encoded into a part latent code Z_j with encoder $Q_{\varphi_j}(Z_j|\hat{S}_j)$, representing the evidence distribution. We use a continuous normalizing flow [8] (CNF) model to learn part priors $P_{\psi_j}(Z_j)$ with parameters ψ_j. Formally, we have the part stylizer loss $\ell_{\mathbf{Z}}$ given by:

$$\begin{aligned} \ell_{\mathbf{Z}} &= \sum_{j=1}^{m} \mathrm{KL}\left(Q_{\varphi_j}\left(Z_j|\hat{S}_j\right) \| P_{\psi_i}(Z_j)\right) \\ &= -\sum_{j=1}^{m} \mathbb{E}_{z_j \sim Q}\left[\log P_{\psi_j}(z_j)\right] + H\left(Q_{\varphi_j}\left(Z_j|\hat{S}_j\right)\right) \end{aligned} \tag{4}$$

where H is the entropy and $P_{\psi_j}(Z_j)$ is a complex distribution transformed from Gaussian. This is equivalent to maximizing $\mathcal{L}_{\mathbf{Z}}$ in Eq (3). See supplement for more details on CNF.

Transformation Sampler. Given part styles $\boldsymbol{z}$, there are different plausible configurations, *i.e.* multiple sets of transformations, that result in valid shapes. Hence, we cannot simply regress the transformations $\mathbf{T}$ for given part styles $\boldsymbol{z}$, leading us to model a conditional *distribution* of transformations $P(\mathbf{T}|\mathbf{Z})$. Learning this distribution is non-trivial because of three main reasons: (i) we have to learn a diverse set of shape variations captured only by the transformation parameters, *e.g.* a dining chair and a bench can have the same set of part styles, (ii) each training example $S \in \mathbf{S}$ only provides one-to-one pairs $(\boldsymbol{z}, \boldsymbol{\tau})$ of part styles and transformations, and (iii) the desired conditional distribution $P(\mathbf{T}|\mathbf{Z})$ may be *multimodal*.

To satisfy these properties, we leverage on conditional Implicit Maximum Likelihood (cIMLE) [24] that trains an implicit generative model, $P_\theta(\mathbf{T}|\mathbf{Z})$ in our case, by encouraging *some* generated output to match the observation from S, in contrast to unimodal approaches [30, 38] that enforces *all* generated outputs to match the observation leading to mode collapse. Concretely, transformation sampler T_θ outputs samples $\boldsymbol{\tau}_k = T_\theta(\boldsymbol{z}, y_k)$ for part style latents $\boldsymbol{z}$ and random latent variable $y_k \sim \mathcal{N}(0, I), y \in \mathbb{R}^{D_\tau}$. We sample multiple latents $y_1, ..., y_K$ and encourage that at least one of them matches the observed data S. As shown in IMLE [23], maximizing the likelihood is then equivalent to minimizing the loss:

$$\ell_{\boldsymbol{\tau}} = \sum_{S \in \mathbf{S}} \min_{k=1,\ldots,K} \ell_{\text{fit}}\left(T_\theta(\boldsymbol{z}_S, y_k), \boldsymbol{\tau}_S\right), \tag{5}$$

where $\boldsymbol{\tau}_S$ is the observed part transformations for shape S. We define ℓ_{fit} as the distance between the generated set $\{\boldsymbol{\tau}_k\}$ and observation $\boldsymbol{\tau}_S$ summed across all parts, given as

$$\ell_{\text{fit}}(\boldsymbol{\tau}, \boldsymbol{\tau}_S) = \sum_{j=1}^{m} \|c_j - c_{S,j}\|_2^2 + \|\log s_j - \log s_{S,j}\|_2^2.$$

Minimizing $\ell_{\boldsymbol{\tau}}$ is equivalent to maximizing $\mathcal{L}_{\boldsymbol{\tau}}$ in Eq (3).

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:30:43 UTC from IEEE Xplore. Restrictions apply.

4.3. Cross Diffusion Network

In order to capture plausible and holistic shapes given our proposed factorization, we model the conditional shape distribution $P(\mathbf{S}|\mathbf{Z},\mathbf{T})$ given part style latent codes and part transformations with our *cross diffusion network*. Specifically, we represent a shape as a distribution of points on its surface. Given segmented shapes S with M parts , we model $P(x|\mathbf{Z},\mathbf{T},j)\ \forall x \in \mathbb{R}^3$ as the probability that x lies on the surface of part S_j. Since we use part semantic labels as an additional condition, this also allows us to output segmented shapes by specifying j. Each point is treated as an independent sample (denoted by the random variable X), which leads to the conditional likelihood of a shape S:

$$P\left(S|\mathbf{Z},\mathbf{T}\right) = \prod_{j=1}^{m} \prod_{x \in S_j} P\left(x|\boldsymbol{z},\boldsymbol{\tau},j\right)^{1}. \tag{6}$$

Our cross diffusion network leverages on denoising diffusion probabilistic model (DDPM) to learn the conditional likelihood $P_\phi(\mathbf{Z},\mathbf{T},j)$ through an iterative denoising process. DDPM models a probability distribution using a reverse process which is a Markov chain with a fixed prior, and to learn DDPM, we approximately maximize the likelihood with the forward process as the approximate posterior. Instead of directly maximizing each conditional likelihood [28, 16], we reparameterize learn ε_ϕ that predicts the noise at a timestep t given the noisy data point $x^{(t)}$ and minimize the distance between the predicted noise ε_ϕ and the ground truth noise ε. Thus maximizing $\mathcal{L}_{\text{recon}}$ in Eq (3) is equivalent to minimizing

$$\ell_{\text{cross}} = \sum_{j=1}^{m} \sum_{x \in S_j} \mathbb{E}_{\varepsilon,\boldsymbol{z},t^{2}} \left[\left\| \varepsilon - \varepsilon_\phi \left(x^{(t)}, \boldsymbol{z}, \boldsymbol{\tau}_S, j, t \right) \right\|_2^2 \right], \tag{7}$$

where the predicted noise ε_ϕ is conditioned on part styles $\boldsymbol{z}$, transformations $\boldsymbol{\tau}$, semantic label j and timestep t. Our cross diffusion network uses our introduced *generalized* forward kernel that allows for the preservation of information from part transformations ($\tau_j = (c_j, s_j)$) in the forward diffusion process while only adding noise to the canonicalized part geometries, *i.e*. part styles. We also show the reverse process of our generalized forward kernel can also be similarly reparameterized arriving at the same loss in Eq (7). Fig. 2 shows one example for the forward diffusion process. Our experiments show that this modification allows for better reconstruction quality and part transformation extrapolation.

We use a cross-attention network with L cross attention layers to instantiate ε_ϕ. For a timestep t, the network predicts $x^{(t-1)}$ conditioned on $(x^{(t)}, j, \tau_j)$. The input to each cross attention layer attends to m tokens each being the concatenation of (z_j, τ_j, j, t), for $j = 1, ..., m$. This design allows a point $x^{(t)}$ to be informed of both the global shape through the m tokens and the local part by concatenating the coordinate $x^{(t)}$ with its corresponding part label and transformation, enabling us to capture and generate more plausible and holistic shapes.

Generalized Forward Kernel. Finally, we introduce a generalization of the forward kernel used to add noise ε to the data points x. Existing works [28, 53] use a forward kernel that diffuses all points on the surface to the standard unit Gaussian. We show that our generalized forward kernel is theoretically equivalent to diffusing all points to a *scaled* and *shifted* Gaussian. Our modification allows to incorporate an additional prior (scale and shift) to the forward process. Specifically, for a d dimensional diffusion process, a shift $\mu \in \mathbb{R}^d$ and variance $\Sigma \in \mathbb{R}^{d\times d}$ parameter is incorporated into the forward kernel so that it becomes:

$$\begin{aligned} &Q\left(X^{(t)}|x^{(t-1)},\mu,\Sigma\right) \\ &= \mathcal{N}\left(\sqrt{\alpha_t}x^{(t-1)} + \left(1-\sqrt{\alpha_t}\right)\mu, \left(1-\alpha_t\right)\Sigma\right), \end{aligned} \tag{8}$$

for $t = 1, \ldots, T$. Here α_t's are variance schedule hyperparameters. As $T \to \infty$, we show that the final distribution approaches a parameterized Gaussian with mean μ and variance Σ (see supplement for the full proof):

$$\begin{aligned} Q\left(X^{(T)}|x^{(0)},\mu,\Sigma\right) &= \mathcal{N}\left(\sqrt{\overline{\alpha}_T}x^{(0)}\right. \\ &\left. + \left(1-\sqrt{\overline{\alpha}_T}\right)\mu, \left(1-\overline{\alpha}_T\right)\Sigma\right) \cong \mathcal{N}\left(\mu,\Sigma\right). \end{aligned} \tag{9}$$

Here $\overline{\alpha}_T = \prod_{t=1}^{T} \alpha_t$. Note that the standard forward kernel is a special case of our generalization by setting $\mu = \mathbf{0}$ and $\Sigma = \text{diag}(\mathbf{1})$. For our task, we set $\mu = c_j$ and $\Sigma = \text{Diag}(s_j^2)$ for points $x \in S_j$.

Please see supplement for more details and full derivations. The total loss is then $\ell_{\text{total}} = \ell_{\text{recon}} + \lambda_1 \ell_{\mathbf{Z}} + \lambda_2 \ell_{\boldsymbol{\tau}}$.

4.4. Enabling Shape Editing

Our method does not only allow for shape generation through *sampling* from individual part style $P(Z_j)$ and transformation $P(\mathbf{T}|\mathbf{Z})$ distributions, but it is also able to *encode* specified parts and *modify* them, allowing for local shape editing and controllable variation synthesis. We highlight that enabling local edits/variations is non-trivial as there is a tension between *preservation* and *adaptation* post-edit. In other words, an ideal edit would keep the unmodified parts unchanged as much as possible, but at the same time still adapt the changes required to maintain a plausible shape after incorporating the specified edit.

1) In the simplest setting, our framework can be trained as an *autoencoder* by deterministically regressing each part latent Z_j and directly feeding the part transformations $\boldsymbol{\tau}_S$ for a shape S. The training objective is simply equivalent

[1] By the same assumption of a deterministic mapping between x and j.

[2] $\varepsilon \sim \mathcal{N}\left(\mathbf{0},\boldsymbol{I}\right), \boldsymbol{z} \sim Q_{\boldsymbol{\varphi}}\left(\boldsymbol{Z}|\hat{S}\right)$, and $t \sim \text{Uniform}\left\{1,\ldots,T\right\}$

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:30:43 UTC from IEEE Xplore. Restrictions apply.

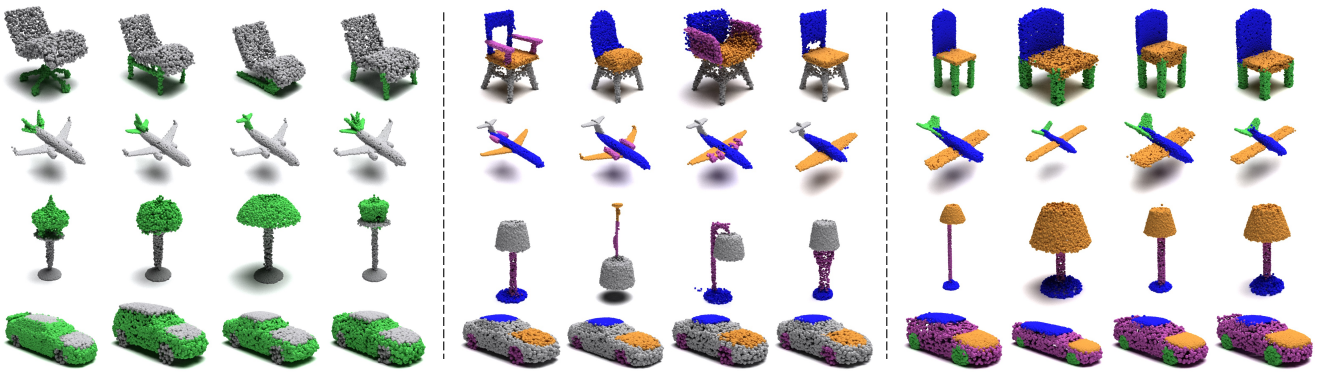

Figure 3: **Generated Shapes with Controlled Variation**. (Left) Re-sampling a selected part style while keeping the rest fixed. (Middle) Fixing a selected part while re-sampling the rest. (Right) Generating multiple part configurations for a given set of part styles. Gray refers to the fixed part while colored parts are being modified.

to ℓ_{recon}. 2) A step further from the autoencoder set-up is to *synthesize variations* of a shape by training the transformation sampler. Given S, we deterministically encode the part styles, and then either a) sample different transformations $\boldsymbol{\tau}$, or b) manually edit selected part transformations τ_j from τ_S, achieving variations on (part) transformation configurations while keeping part styles fixed. This set-up also allows for shape mixing-and-matching with transformation variations. 3) Moreover, we also enable *local* editing where specified part j of a shape can change while keeping the rest fixed that is done by either a) resampling the corresponding part latent code from $P(Z_j)$ or b) interpolating between part styles. The applications are shown in the results section.

5. Results

5.1. Dataset and Evaluation Protocol

We use four classes from ShapeNet [6] dataset: chair, airplane, lamps, and cars. We train/test the networks per object class with the splits provided by [49]. Each category contains 3053, 2349, 1261, 740 training shapes and 704, 341, 286, 158 test shapes, respectively. The semantic labels for all classes come from [49]. See supplement for implementation details — network architecture, training time, etc.

For our task on controllable part-based generation, we propose to measure intra-part and inter-part level scores. For more details, please refer to the supplement.

Intra-part Score. We evaluate the quality of individual part distributions $P(S_j) \forall j$ using the standard generation metrics following [46]: minimum matching distance (MMD-P), coverage (COV-P) and 1-NN classifier accuracy[3](1NNA-P), measuring the similarity between the distributions of canonicalized parts of the generated shapes compared to the test set from [49] of segmented shapes, where the score is computed for each part and averaged across all *P*arts (hence the suffix '-P').

Chair	MMD-P (↓)	COV-P (↑)	1NNA-P
PointFlow [46]	4.68	27.3	87.77
DPM [28]	4.17	28.2	85.65
ShapeGF [5]	3.52	42.3	68.65
LION [52]	3.99	35.1	69.25
DiffFacto (Ours)	**3.27**	**42.5**	**65.23**

Airplane	MMD-P (↓)	COV-P (↑)	1NNA-P
PointFlow [46]	4.61	32.0	86.11
DPM [28]	3.52	37.7	78.74
ShapeGF [5]	3.50	40.0	72.04
LION [52]	3.68	38.8	68.73
DiffFacto (Ours)	**3.20**	**46.2**	**68.72**

Table 1: **Global Shape Code Baselines.** MMD-P score is multiplied by 10^{-2}. COV-P and 1NNA-P are reported in %.

Inter-part Score. We also measure the local part-to-part coherence of the generated shape as having control in individual parts may result in disjoint or incoherent outputs. We use a snapping metric (SNAP) that measures local connectivity between independently generated parts, given as the Chamfer distance (CD) between the N_{snap}-closest points between connected parts. We report the average score across all connections of all generated samples.

Plausibility. We also further evaluate the plausibility of our generated shapes by using them to augment training datasets for point cloud segmentation. The idea is if our approach generates *novel* and coherent shapes with part labels then using them for data augmentation would improve the segmentation score of part segmentation networks.

Human Study. We also conduct a user study to evaluate

[3]Arrow for 1NNA-P is left out in the table because an optimal score for this metric is 50 %.

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:30:43 UTC from IEEE Xplore. Restrictions apply.

	Ctrl-ShapeGF	Ctrl-LION	**DiffFacto** (Ours)
SNAP (↓)	41.12	31.76	**13.32**

Table 2: **Control-Enabled Baselines.** Snapping metric on three connections for chairs: back to leg or seat; seat to legs; and arms to seat or back. (CD $\times 10^{-2}$).

controllability, methods being evaluated each generate a triplet of edited shapes, and users are asked to select the triplet containing shapes that are most plausible, where an 'abstain' option can be selected. An edit for a given shape is defined as resampling a pre-selected parts to output a novel shape where plausibility is measured.

5.2. Baselines

Global Shape Code. Existing point cloud generation works do not explicitly model individual parts as they sample from a single *global* shape distribution, which *do not* provide individual part-level control. To evaluate individual part distributions and measure the intra-part score, we use a pre-trained part segmenter [34] to decompose generated shapes into parts. We compare against recent works PointFlow [46], DPM [28], ShapeGF [5] and LION [52] on intra-part level scores. Inter-part scores are not measured as these works directly generate a full shape.

Control-Enabled. We also introduce new baselines that allow for part-level control. We modify ShapeGF [5] and LION [52] to have part-level control (prefixed by "Ctrl-") by modeling independent part distributions, *i.e.* $P(\mathbf{S}) = \prod_j P(S_j|w_j)$, where each part latent distribution is modeled with a (hierarchical) variational encoder $Q(w_j|S_j)$. These baselines allow for independent sampling at the part-level unlike existing approaches. We measure the inter-part score for these baselines to evaluate the coherence of the generated shape. Intra-part scores are not measured as these baselines are individually trained per part.

5.3. Controllable Shape Generation

Fig. 3 shows shapes generated by sampling from different components in our factorization enables both control in part style and part configurations: Our approach is able to *sample* (left) or *fix* (middle) a specified part, and we are also able to generate various plausible configurations of the shape (right) with fixed part styles.

Comparison with Global Shape Code Baselines. Tab. 1 shows the intra-part scores of our approach compared to the global shape baselines, showcasing that our approach learns better individual part-level distributions.

Comparison with Control-Enabled Baselines. Tab. 2 shows the inter-part scores of our approach compared to the control-enabled baselines. We output more coherent shapes than the baselines that naively enable part-level control without the modeling of part relationships

	Orig.	+ Multi (700)	+ Control (60)
PointNet	0.709	**0.788**	0.780
PointNet++	0.800	**0.808**	0.801

Table 3: **Plausibility score.** mIOU on the cars from [49] trained original and augmented datasets.

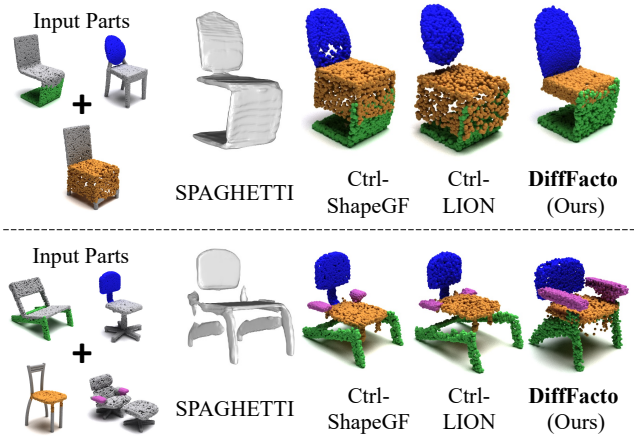

Figure 4: **Part Style Mixing.** The colored parts from the left are selected and mixed to provide the shapes on the right.

Plausibility. We use the car category containing the least training data originally with 704 shapes (Orig.), and augment it with 700 randomly generated shapes by DiffFacto (+ Multi). Moreover, we test our capability for control by augmenting with only 60 (controlled) race cars, with very few examples in the original training data. Tab. 3 shows that in both cases off-the-shelf part segmentation networks improve by augmenting the training set with our generated shapes.

Human Study. Our human study has 100 participants comparing our approach with the control-enabled baselines, *i.e.* Ctrl-ShapeGF and Ctrl-LION. We drew 10 shapes from each of the methods with randomly selected parts to edit, and on average the participants favour **85%** (8.5 out of 10) of our generated shapes more than other baselines.

5.4. Shape Editing

We demonstrate that our controllable generation approach also allows for various shape-editing applications.

Part Style Mixing. We showcase our ability for part style mixing in Fig. 4. We compare with SPAGHETTI [15], an implicit-based shape editing work, as well as our control-enabled baselines Ctrl-ShapeGF and Ctrl-LION[4]. As shown, when selecting a combination of parts from different shapes, our approach generates a novel and coherent shape from different input parts, compared to other approaches that are unable to adapt the parts to produce a plausible output.

[4]We note that SPAGHETTI [15] requires mesh supervision.

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:30:43 UTC from IEEE Xplore. Restrictions apply.

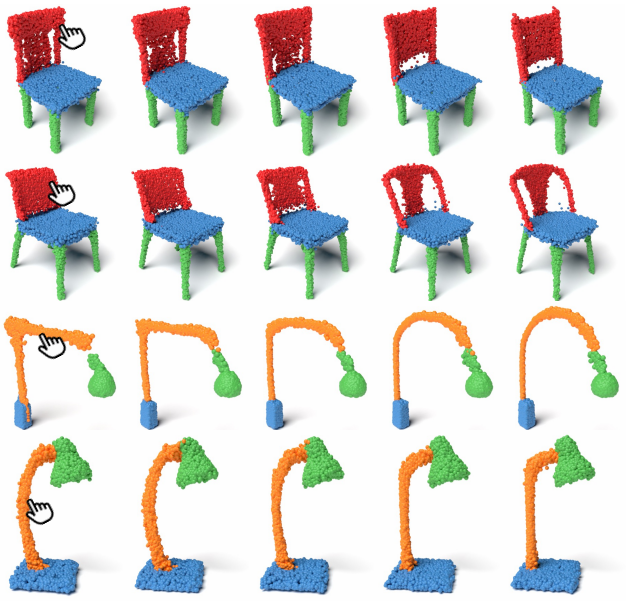

Figure 5: **Part Interpolation**. We interpolate the chair backs (red) in the first 2 rows and the lamp poles (orange) in the last 2 rows (indicated by the hand icons).

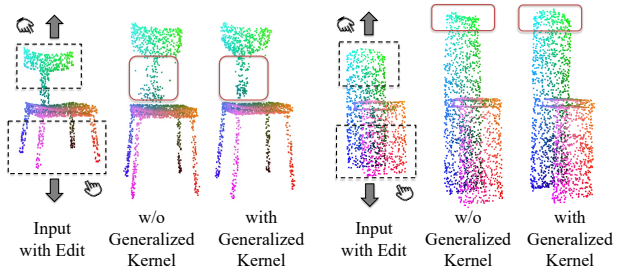

Figure 6: **Part Editing.** In both examples the user stretches the lengths of the chair backs and legs by modifying the corresponding transformations.

Part-level Interpolation. Fig. 5 qualitatively shows our part-level interpolation performance, where for each example we interpolate only one selected part latent z_j. Thanks to our factorized probabilistic formulation, we are able to interpolate only the selected part while keeping the geometry of the other parts unchanged. In the meantime, the *transformations* of the other parts are automatically adapted to make the shape globally coherent.

Transformation Editing. Our approach also enables direct user editing on the part transformations τ_j for a selected part S_j. We directly optimize y to find $\boldsymbol{\tau}$ that satisfies the edit while still traversing along the space of valid part configurations. As Fig. 6 (left) shows, elongating the chair back retains its thin geometric structure while still keeping the shape plausibility.

	Separate	*Post Transform*	*Global Agnostic*	**DiffFacto** (Ours)
SNAP (↓)	25.24	18.23	19.29	**13.32**

Table 4: **Ablation on our Factorization.** Snapping metric on three connections for chairs: back to leg or seat; seat to legs; and arms to seat or back. (CD $\times 10^{-2}$).

Direct reg.	cVAE [38]	cGAN [30]	cIMLE [?] (Ours)
13.38	7.33	11.48	**4.97**

Table 5: **Multi-modality of Transformation Sampler.** Shape inversion on the chair category (CD $\times 10^{-4}$).

5.5. Ablations

Factorized Joint Prior. We ablate our joint factorized prior $P(\mathbf{Z}, \mathbf{T})$ by replacing it with separate independent part distributions (*Separate*). Each part distribution $P(S_j)$ is modeled with a separate CNF prior. Tab. 4 shows that our approach achieves better inter-part score.

Cross Diffusion. We also evaluate several variants of our cross diffusion network, termed as follows: *Post Transform* – we remove the cross diffusion network and instead model the conditional likelihood of points only on part styles, then separately applying the part transformations as a post-processing step. *Global Agnostic* – we remove the cross attention component that provides global shape information through the m tokens, and model only the point distribution $P(X|z_j, \tau_j, j)$ for each part. Tab. 4 shows that our approach also achieves better inter-part level score.

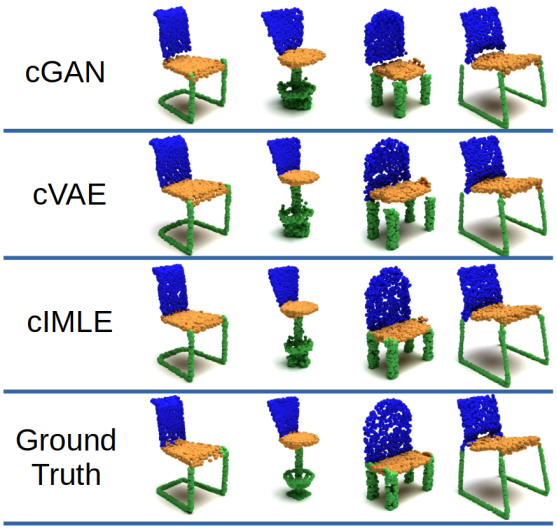

Figure 7: **Shape Inversion Comparison.** We show shape inversion examples using different implicit probabilistic methods to model the transformation distribution given part style latents.

Transformation Sampler. We ablate our transformation

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:30:43 UTC from IEEE Xplore. Restrictions apply.

sampler that uses cIMLE [24] and compare it with direct regression of τ, as well as unimodal cVAE [38] and cGAN [30] on shape inversion. Tab. 5 quantitatively shows that our approach achieves the best results with a large margin and Fig. 7 shows examples of inverted shapes using different implicit methods. Notice that across all examples, cIMLE is able to recover most accurately the correct transformation through inversion because it models a multimodal distribution where different modes can be recovered during the sampling process.

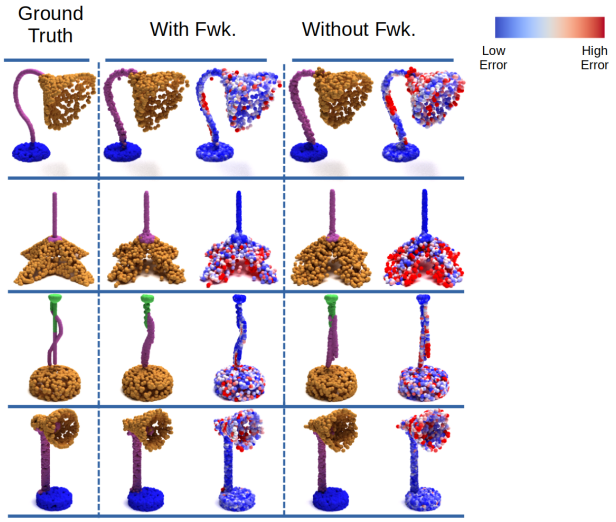

Figure 8: **Qualitative Comparison w. Generalized Forward Kernel.** Examples of reconstructed lamps with our generalized kernel **(With Fwk.)** versus without the generalized kernel **(Without Fwk.)** For each example, we show the reconstructed shape **(Ground Truth)** where each point is colored by the minimum distance to points in the ground truth shape.

Generalized Forward Kernel. Our generalized forward kernel works by modeling the diffusion prior as a transformed Gaussian distribution that captures informative positional and scale information of the part. Fig. 6 shows that such a kernel allows better transformation extrapolation, where geometry is better preserved on extreme user edits. Moreover, Fig. 8 shows qualitative examples of the advantages of using our generalized forward kernel as illustrated by the heat map on the per-point reconstruction error. We note that our generalized forward kernel is able to model complex part geometry better than the standard forward kernel (see the cap on rows 1, 2, and 4 in the figure) because of the additional size and location prior information incorporated into the diffusion process.

6. Future Works and Limitations

Our method by design requires segmented shapes for training, as our cross diffusion network and generalized forward kernel require a hard assignment of points to the semantic part of the shape, *i.e.* it conditions on label j. Although this allows us to learn smooth latent spaces for each part, as well as global shape distribution $P(\mathbf{S})$, it constrains us in training with datasets that are semantically labeled. A future direction would be a formulation that enables soft assignments, which would allow us to train on unsegmented data .

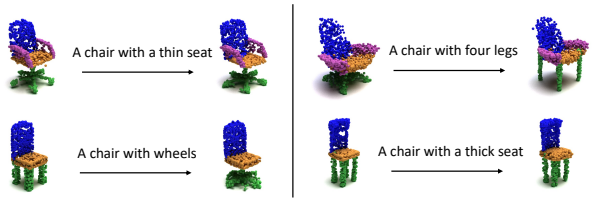

Figure 9: **Potential Language Guided Edits.** We show prototypes of potential future work where we can edit existing shapes via our part style latent distributions with language guidance. The left sides are input shapes and we update the part latent vectors based on the language inputs. The resulting shapes are shown on the right side of the arrows.

Another promising future direction is to enable language-driven editing through our proposed factorization. Our factorization allows for localized edits, where language can be used to identify which semantic part, j, to edit as well as the direction to modify its corresponding latent code, Z_j, to adhere with the language description. As a preliminary demonstration, we use the dataset from ShapeGlot [2] that consists of a triplet of shapes with a corresponding language description. We then select a negative shape and edit it using DiffFacto in order to match with the corresponding text description. A match is enforced by optimizing the edit to match the positive example in the triplet. Fig. 9 shows some visual examples.

7. Conclusion

In this paper we propose DiffFacto, a deep generative probabilistic framework for generating 3D point clouds in a controllable manner. By factorizing out the shape distribution into individual semantically-meaningful parts plus their transformations, we allow for intuitive control over the generated shapes. The framework is also flexible and readily available for various applications ranging from style mixing to configuration editing.

Acknowledgements. This work is supported by ARL grant W911NF-21-2-0104, a Vannevar Bush Faculty Fellowship, the Natural Science Foundation of China (Project No. 62220106003), Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology, and the Natural Sciences and Engineering Research Council of Canada. We very much appreciate Congyue Deng for her helpful discussion at the early stage of this project. We are also grateful for the advice and help from Colton Stearns and Davis Rempe.

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:30:43 UTC from IEEE Xplore. Restrictions apply.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. 2

[2] Panos Achlioptas, Judy Fan, Robert X. D. Hawkins, Noah D. Goodman, and Leonidas J. Guibas. Shapeglot: Learning language for shape differentiation. *CoRR*, abs/1905.02925, 2019. 9

[3] Péter Borosán, Reid Howard, Shaoting Zhang, and Andrew Nealen. Hybrid mesh editing. In *Eurographics (short papers)*, pages 41–44, 2010. 2

[4] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015. 2

[5] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *ECCV*, 2020. 1, 2, 3, 6, 7

[6] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 6

[7] Siddhartha Chaudhuri, Daniel Ritchie, Jiajun Wu, Kai Xu, and Hao Zhang. Learning generative models of 3d structures. In *Computer Graphics Forum*, volume 39, pages 643–666. Wiley Online Library, 2020. 2

[8] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018. 4

[9] Fabrizio Corda, Jean-Marc Thiery, Marco Livesu, Enrico Puppo, Tamy Boubekeur, and Riccardo Scateni. Real-time deformation with coupled cages and skeletons. In *Computer Graphics Forum*, volume 39, pages 19–32. Wiley Online Library, 2020. 2

[10] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 2

[11] Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao(Richard) Zhang. SDM-NET: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2019)*, 38(6):243:1–243:15, 2019. 2

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[13] Zekun Hao, Hadar Averbuch-Elor, Noah Snavely, and Serge Belongie. Dualsdf: Semantic shape manipulation using a two-level representation. pages 7631–7641, 2020. 2

[14] Amir Hertz, Or Perel, Raja Giryes, Olga Sorkine-Hornung, and Daniel Cohen-Or. Spaghetti: Editing implicit shapes through part aware generation. *ACM Transactions on Graphics (TOG)*, 41(4):1–20, 2022. 2

[15] Amir Hertz, Or Perel, Raja Giryes, Olga Sorkine-Hornung, and Daniel Cohen-Or. Spaghetti: Editing implicit shapes through part aware generation. In *SIGGRAPH Asia*, 2022. 3, 7

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 2, 5

[17] Ka-Hei Hui, Ruihui Li, Jingyu Hu, and Chi-Wing Fu. Neural wavelet-domain diffusion for 3d shape generation, 2022. 2

[18] Hyeongju Kim, Hyeonseung Lee, Woo Hyun Kang, Joun Yeop Lee, and Nam Soo Kim. Softflow: Probabilistic framework for normalizing flow on manifolds. *Advances in Neural Information Processing Systems*, 33:16388–16397, 2020. 2

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013. 2

[20] Juil Koo, Ian Huang, Panos Achlioptas, Leonidas J Guibas, and Minhyuk Sung. Partglot: Learning shape part segmentation from language reference games. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16505–16514, 2022. 2

[21] Jun Li, Chengjie Niu, and Kai Xu. Learning part generation and assembly for structure-aware shape synthesis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11362–11369, 2020. 2

[22] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017. 2

[23] Ke Li and Jitendra Malik. Implicit maximum likelihood estimation. *arXiv preprint arXiv:1809.09087*, 2018. 4

[24] Ke Li, Shichong Peng, Tianhao Zhang, and Jitendra Malik. Multimodal image synthesis with conditional implicit maximum likelihood estimation. *International Journal of Computer Vision*, 128:2607–2628, 2020. 2, 4, 9

[25] Ruihui Li, Xianzhi Li, Ka-Hei Hui, and Chi-Wing Fu. Sp-gan: Sphere-guided 3d shape generation and manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–12, 2021. 2

[26] Shidi Li, Miaomiao Liu, and Christian Walder. Editvae: Unsupervised part-aware controllable 3d point cloud shape generation, 2021. 2

[27] Connor Z. Lin, Niloy J. Mitra, Gordon Wetzstein, Leonidas Guibas, and Paul Guerrero. Neuform: Adaptive overfitting for neural shape editing, 2022. 2

[28] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *CVPR*, June 2021. 1, 2, 3, 5, 6, 7

[29] Zhaoyang Lyu, Zhifeng Kong, Xudong Xu, Liang Pan, and Dahua Lin. A conditional point diffusion-refinement paradigm for 3d point cloud completion. *arXiv preprint arXiv:2112.03530*, 2021. 2

[30] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014. 2, 4, 8, 9

[31] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019. 3

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:30:43 UTC from IEEE Xplore. Restrictions apply.

[32] Charlie Nash and Christopher KI Williams. The shape variational autoencoder: A deep generative model of part-segmented 3d objects. In *Computer Graphics Forum*, volume 36, pages 1–12. Wiley Online Library, 2017. 2

[33] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 2

[34] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 7

[35] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015. 2

[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2

[37] Thomas W Sederberg and Scott R Parry. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 151–160, 1986. 2

[38] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015. 2, 4, 8, 9

[39] Yongbin Sun, Yue Wang, Ziwei Liu, Joshua Siegel, and Sanjay Sarma. Pointgrow: Autoregressively learned point cloud generation with self-attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 61–70, 2020. 2

[40] Minhyuk Sung, Zhenyu Jiang, Panos Achlioptas, Niloy J Mitra, and Leonidas J Guibas. Deformsyncnet: Deformation transfer via synchronized shape deformation spaces. *arXiv preprint arXiv:2009.01456*, 2020. 2

[41] Minhyuk Sung, Hao Su, Vladimir Kim, Siddhartha Chaudhuri, and Leonidas Guibas. ComplementMe: Weakly-supervised component suggestions for 3d modeling. *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia)*, 2017. 1

[42] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in neural information processing systems*, 33:19667–19679, 2020. 2

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[44] Kai Wang, Yu-An Lin, Ben Weissmann, Manolis Savva, Angel X Chang, and Daniel Ritchie. Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019. 2

[45] Qun-Ce Xu, Tai-Jiang Mu, and Yongliang Yang. A survey of deep learning-based 3d shape generation. *Computational Visual Media*, 2022. 1

[46] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *ICCV*, 2019. 1, 2, 3, 6, 7

[47] Jie Yang, Kaichun Mo, Yu-Kun Lai, Leonidas J Guibas, and Lin Gao. Dsg-net: Learning disentangled structure and geometry for 3d shape generation. *ACM Transactions on Graphics (TOG)*, 42(1):1–17, 2022. 2

[48] Ximing Yang, Yuan Wu, Kaiyi Zhang, and Cheng Jin. Cpcgan: A controllable 3d point cloud generative adversarial network with semantic label generating. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3154–3162, 2021. 2

[49] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *SIGGRAPH Asia*, 2016. 6, 7

[50] Kangxue Yin, Zhiqin Chen, Siddhartha Chaudhuri, Matthew Fisher, Vladimir Kim, and Hao Zhang. Coalesce: Component assembly by learning to synthesize connections. In *3DV*, 2020. 1, 2

[51] Shin Yoshizawa, Alexander G Belyaev, and Hans-Peter Seidel. Free-form skeleton-driven mesh deformations. In *Proceedings of the eighth ACM symposium on Solid modeling and applications*, pages 247–253, 2003. 2

[52] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. In *NeurIPS*, 2022. 1, 2, 6, 7

[53] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5826–5835, October 2021. 2, 5

[54] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through voxel diffusion. In *ICCV*, pages 5826–5835, October 2021. 2

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:30:43 UTC from IEEE Xplore. Restrictions apply.