Pacific Graphics 2025
M. Christie, N. Pietroni, and Y.-S. Wang
(Guest Editors)

Volume 44 (2025), Number 7

SPLICE: Part-Level 3D Shape Editing from Local Semantic Extraction to Global Neural Mixing

Submission ID: 1159

1. Implementation Details

Part Feature Extraction. Each part is first converted into a binary voxel grid by applying Poisson Surface Reconstruction (PSR) to a ShapeNet point cloud and thresholding the resulting SDF values. This binary representation effectively masks connectivity information at cut boundaries: SDF-based inputs would retain high distance values across cuts, biasing the model toward original connections.

We then transform the dense grid into a sparse voxel representation and apply a sequence of 3×3×3 convolutions and max-pooling layers. Feature channels increase from $1 \rightarrow 8$, doubling at each pooling until reaching 512 channels. The network branches into two heads: a 10-dimensional pose head (for Gaussian parameters) and a 512-dimensional shape code head. Both heads are implemented with 1×1×1 convolutions.

Pose Feature Mixing. For each edited Gaussian proxy, we extract the six ellipsoid endpoints along its principal axes at radius 1.75. We encode these 18 coordinates with a SIREN-based positional mapping $\phi(\cdot)$, then concatenate the result with the 512-dimensional shape code and pass through a four-layer MLP (with skip connections and layer normalization) to produce the 256-dimensional part embedding. We adopt random affine perturbations during training: translations sampled uniformly in [–0.5, 0.5], scales as $\exp(\mathcal{U}(-0.5, 0.5))$, and rotations in $(-\pi/4, \pi/4)$ applied with 10% probability to avoid ambiguity in principal axis orientation.

Attention-based Shape Decoding. A query point $\mathbf{x} \in \mathbb{R}^3$ is first encoded by the same SIREN mapping $\phi(\mathbf{x})$. We then use a transformer decoder without self-attention, so that points remain independent, to attend over the set of part embeddings and produce a 128-dimensional local feature $\mathbf{h}_x$. Concatenating $\phi(\mathbf{x})$ and $\mathbf{h}_x$, we decode occupancy via a NeRF-style 4-layer MLP.

Diffusion model. We trained the diffusion module following the architectural and training configurations of 3DShape2VecSet [ZTNW23] and Improved Denoising Diffusion Probabilistic Models [ND21].

To stabilize optimization, we applied rescaling to different components of the concatenated input vector ($1+15+256$ dimensions) in order to balance their relative magnitudes. The binary validity flag was scaled into the range $[-0.01, 0.01]$, ensuring that it is only determined at the final stage without interfering with the denoising process. For the Gaussian embedding $\mathbf{g}_i$, we multiplied the positional and ellipsoidal radius components by 5, while keeping the rotation matrix unchanged, since the unit-norm vectors of the rotation matrix naturally span a larger range. We further applied a random scaling factor between 0.8 and 1.2 to the input shapes to improve robustness. For the shape code $\mathbf{z}_i$, we applied a scale of 0.1, which keeps $\mathbf{z}_i$ relatively blurred during the denoising of $\mathbf{g}_i$. This prevents the model from overfitting to a fixed correspondence between $\mathbf{g}_i$ and $\mathbf{z}_i$, and we found this adjustment to be crucial for achieving high-quality results. All rescaled values are reverted to their original ranges before being fed into the decoder.

We used the standard 1000-step diffusion schedule, where generating a complete set of parts takes approximately 5 seconds. For refinement tasks, we adopt a shortened sequence consisting of 100 noising steps followed by 100 denoising steps.

Training and Inference. During training, we sample 3,000 points uniformly inside the shape's bounding box and add Gaussian noise ($\sigma = 0.02$) around the surface, then resample another 3,000 points to balance interior/exterior supervision. At inference time, we generate a 128^3 grid of query points and process them in batches of 1,000,000 through the decoding network, computing occupancy values in approximately 1 s per shape. Performance can be further improved by coarse-to-fine sampling strategies around predicted surfaces.

2. More Random Generation Result

Due to space limitations, we present only a small portion of the randomly generated results in the main text. In Fig. 2, additional generated results are provided. The shapes produced through our shape representation training exhibit significant structural diversity.

3. Structure-Preserving Shape Generation

Our diffusion generative model is capable of generating a wide variety of shapes conditioned on a given structure. As illustrated in Fig. 1, several examples of such results are presented. Additionally, we provide a preview of the corresponding part-structure representation $\mathbf{g}$ for each generated shape.

4. Editing and Mixing Shapes from Scanned Objects

We utilized the Apple iPhone 16 Pro to scan several types of chairs. Our method requires only a few annotations of bounding boxes to

import scanned real-world shapes. As shown in Fig. 3, our approach effectively demonstrates the editing of real-world shapes. Additionally, it can address and partially rectify imperfections in scanning.

References

[ND21] NICHOL A., DHARIWAL P.: Improved denoising diffusion probabilistic models. *ArXiv abs/2102.09672* (2021). 1

[ZTNW23] ZHANG B., TANG J., NIESSNER M., WONKA P.: 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics (TOG) 42* (2023), 1 – 16. 1

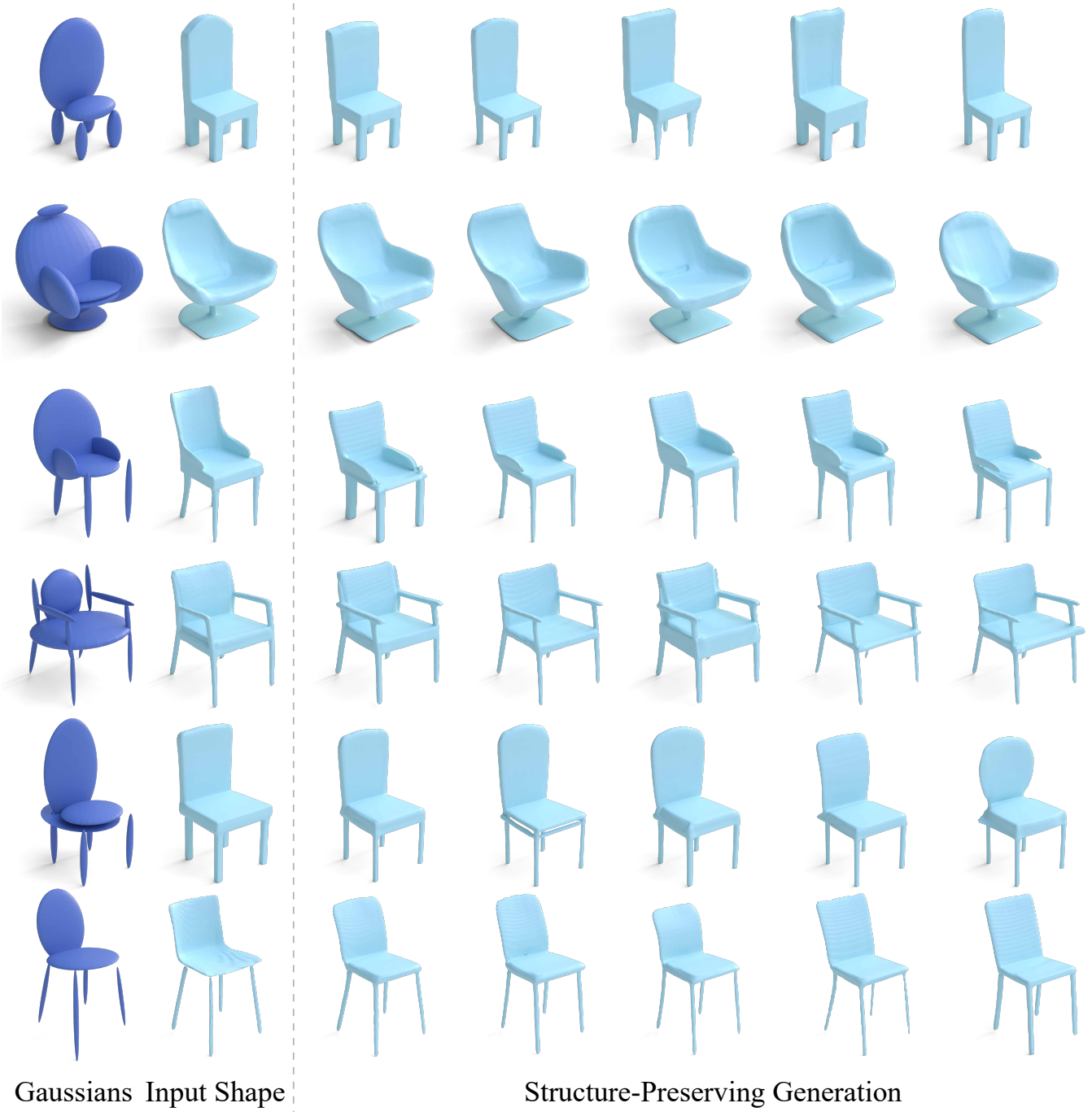

Figure 1: *Visual results of structure-preserving shape generation.*

Figure 2: *More visual results of shape random generation.*

Figure 3: *Visual results of editing and mixing shapes from scanned objects.*