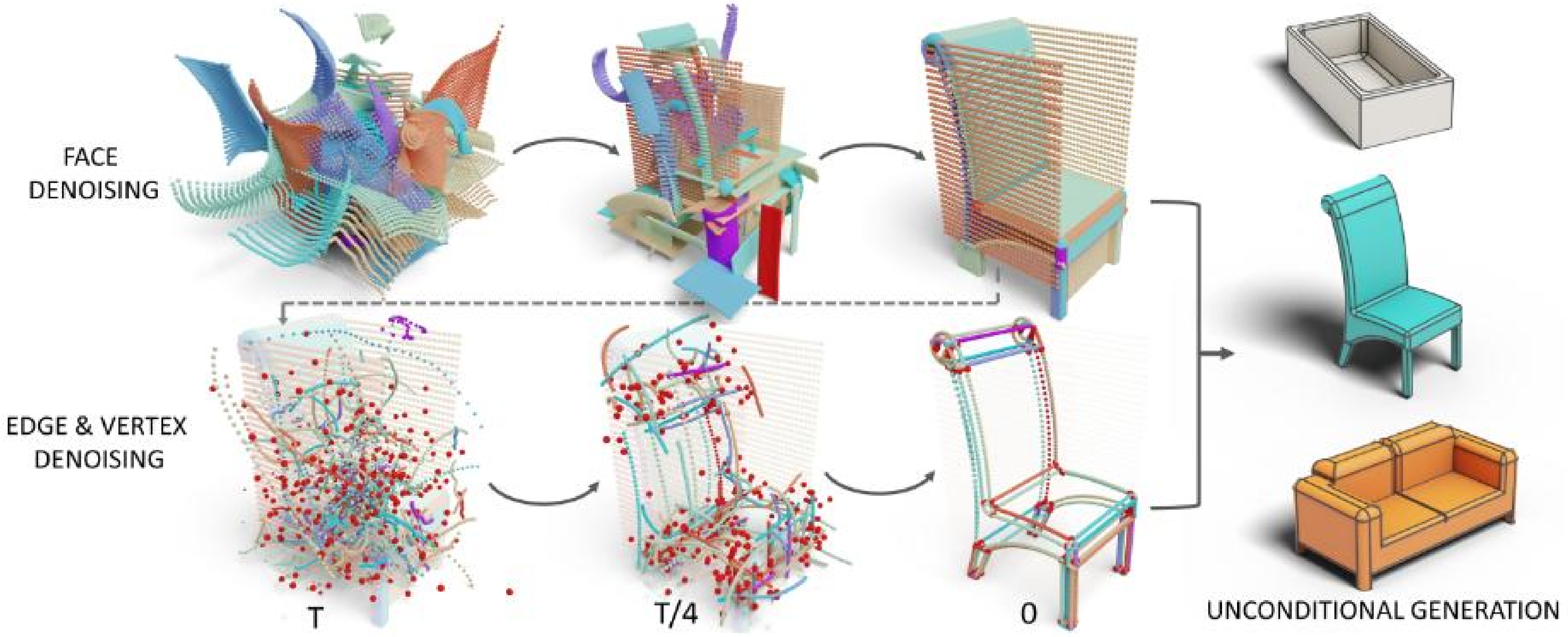
BrepGen: A B-rep Generative Diffusion Model with Structured Latent Geometry

Xiang Xu [1,2] Joseph G. Lambourne[2] Pradeep Kumar Jayaraman[2]
Zhengqing Wang[1] Karl D.D. Willis[2] Yasutaka Furukawa[1]
[1]Simon Fraser University [2]Autodesk Research

https://arxiv.org/abs/2401.15563

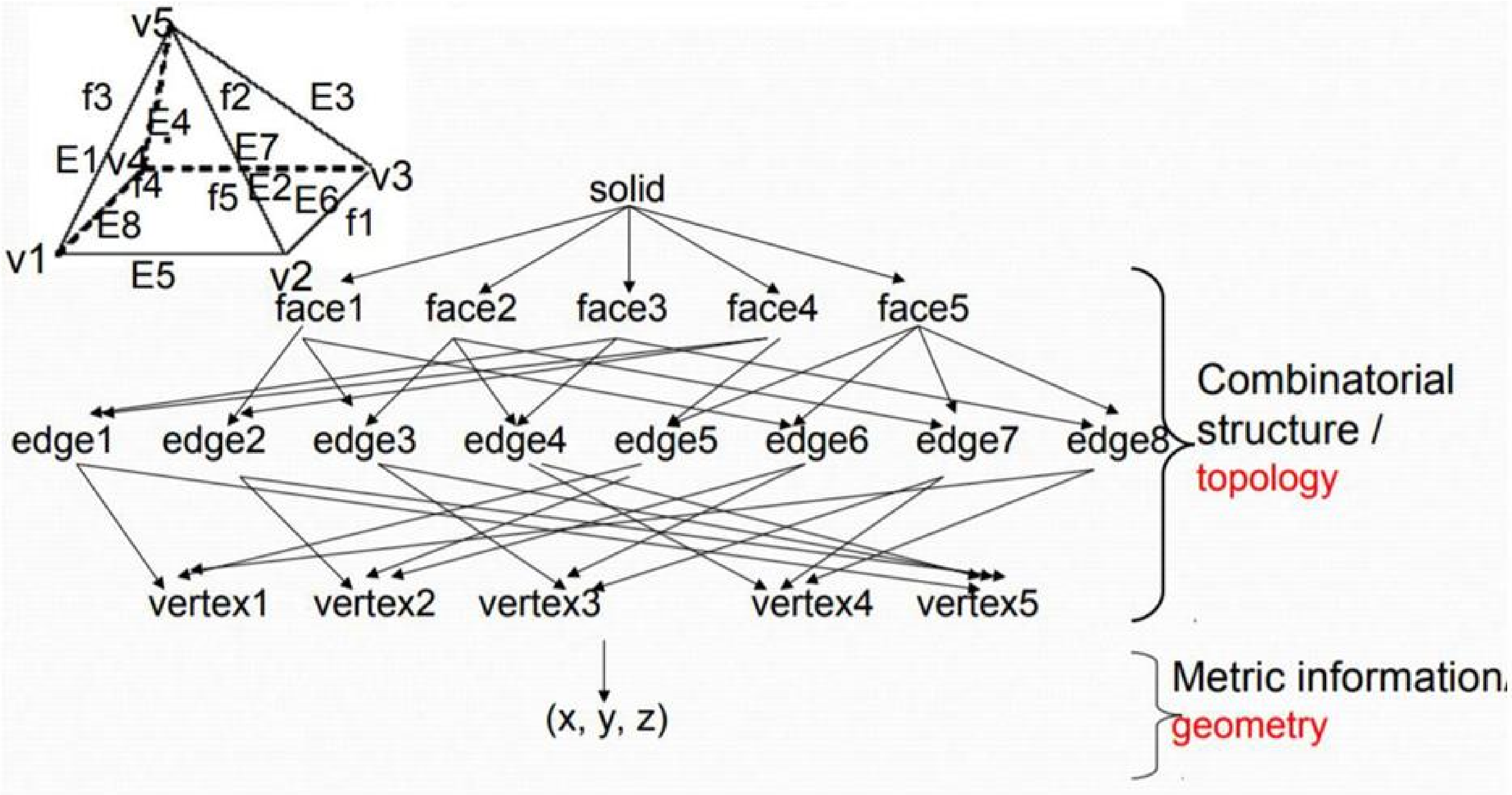
Introduction

This paper presents BrepGen, a diffusion-based generative approach that directly outputs a Boundary representation (B-rep) Computer-Aided Design (CAD) model.



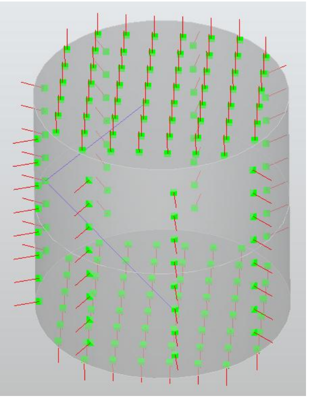
Background

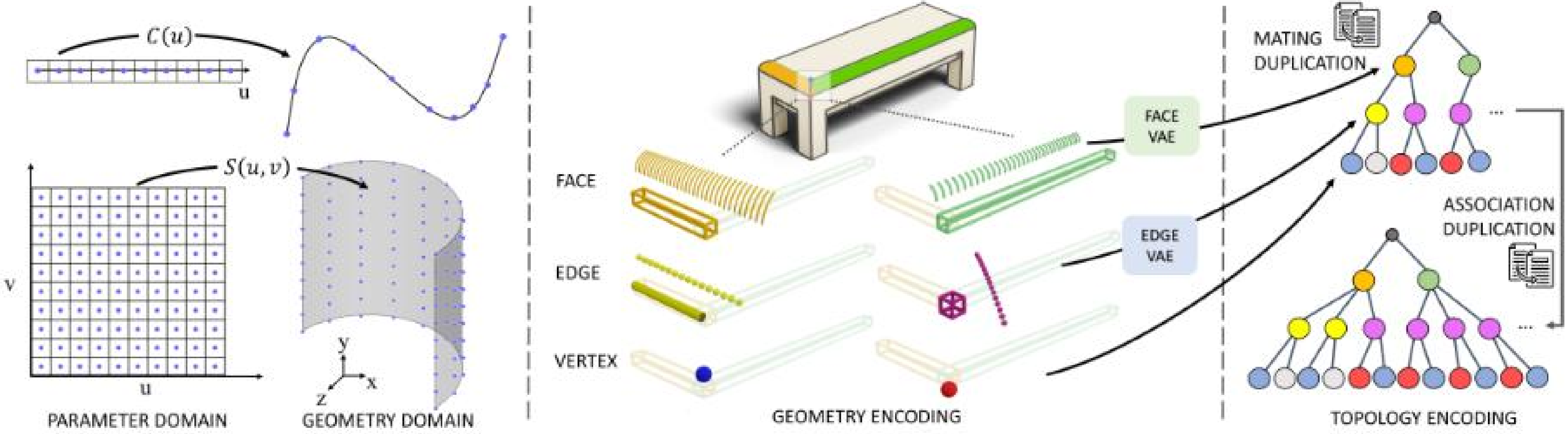
A **CAD model** represents a complex object in a format that permits editing in CAD software. **Boundary representation (B-REP)** is one such representation; it involves analytic **faces**, **edges**, and **vertices**, and explicit topological relationships between them.



Structured Latent Geometry

A face, an edge, or a vertex node encodes its geometry information into a node feature with **1)** a global position as bounding box parameters or a point and **2)** local shape details as a latent code.

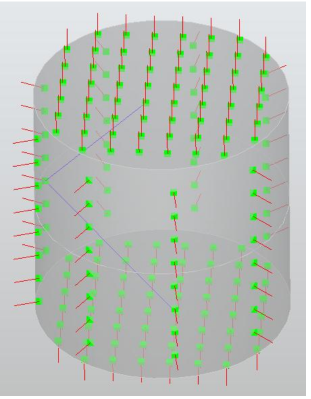


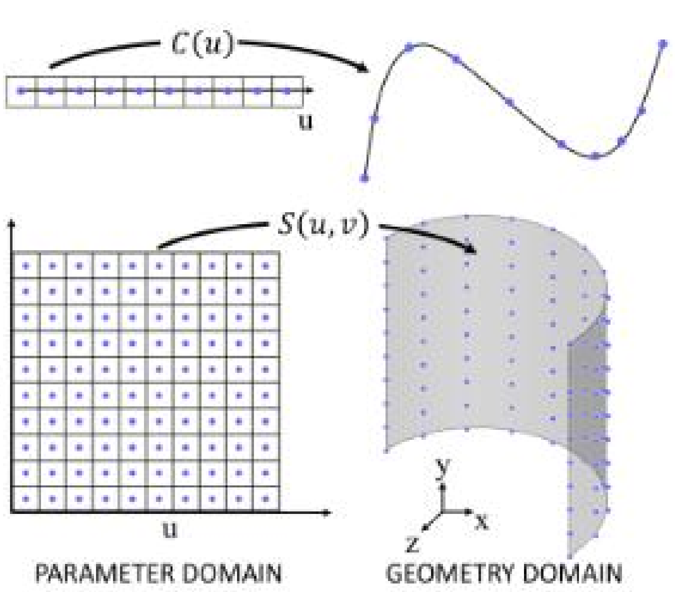


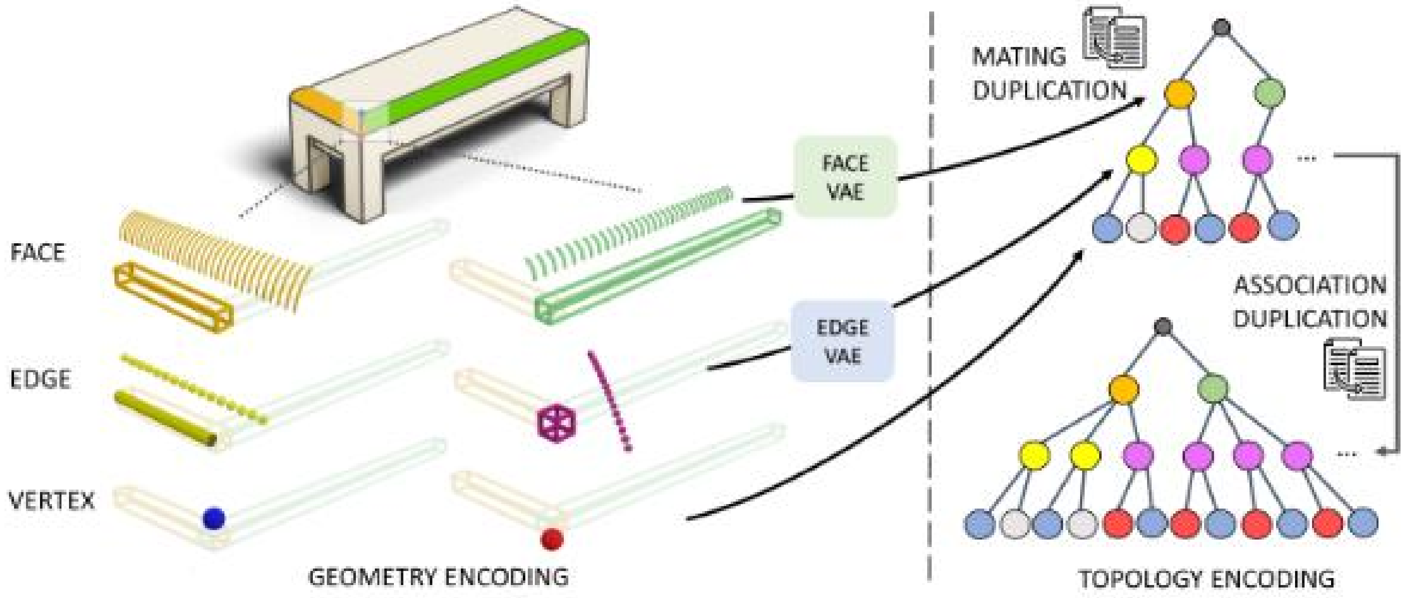
Structured Latent Geometry Face

The shape feature F_S is a 2D array of 3D points sampled on the parametric surface, and **the position feature** F_p is the axis-aligned bounding box enclosing the points: $F_p = [x_1, y_1, z_1, x_2, y_2, z_2]$ encoding the bottom-left and top-right corners.

The feature of a face node is then defined as $F = [F_s, F_p]$





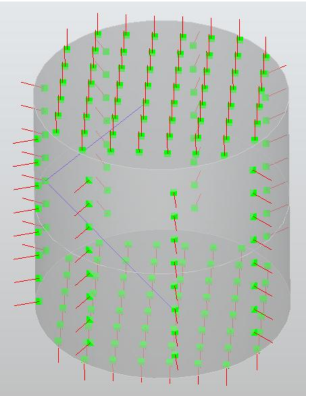


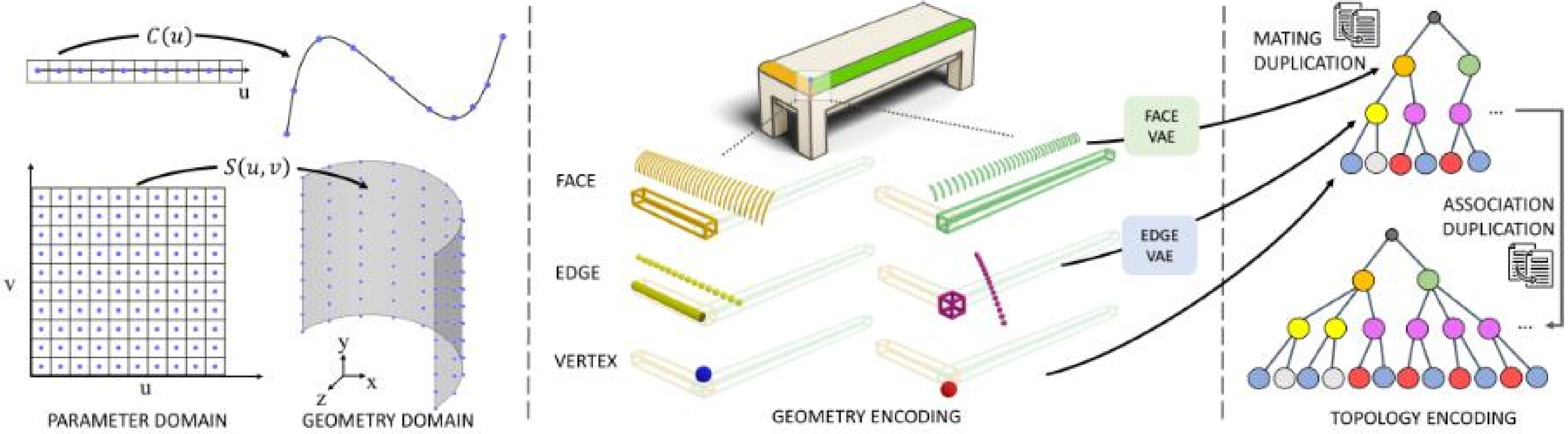
Structured Latent Geometry

Edge, Vertex

The shape feature E_S is a 1D array of 3D points sampled along the parametric curve, and **the position feature** E_p is again defined as the bounding box parameters enclosing the points.

The feature of a vertex node is its point coordinate $V = (x, y, z)$.



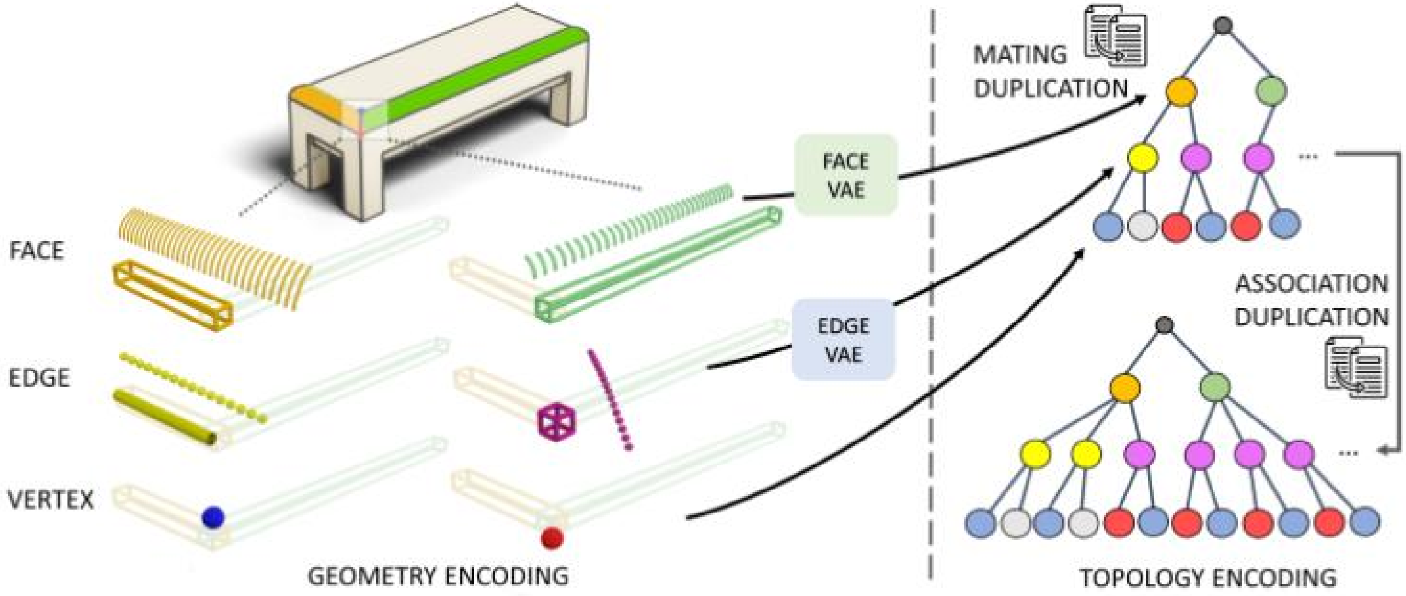


Topology Encoding by Node Duplication

Mating Duplication

A shared geometric element (i.e., an edge shared by faces or a vertex shared by edges) is duplicated, which creates one copy under each parent and turns a B-rep graph structure into a tree.

Mating relations can be later recovered by merging edge or vertex nodes with similar geometry across different parents.

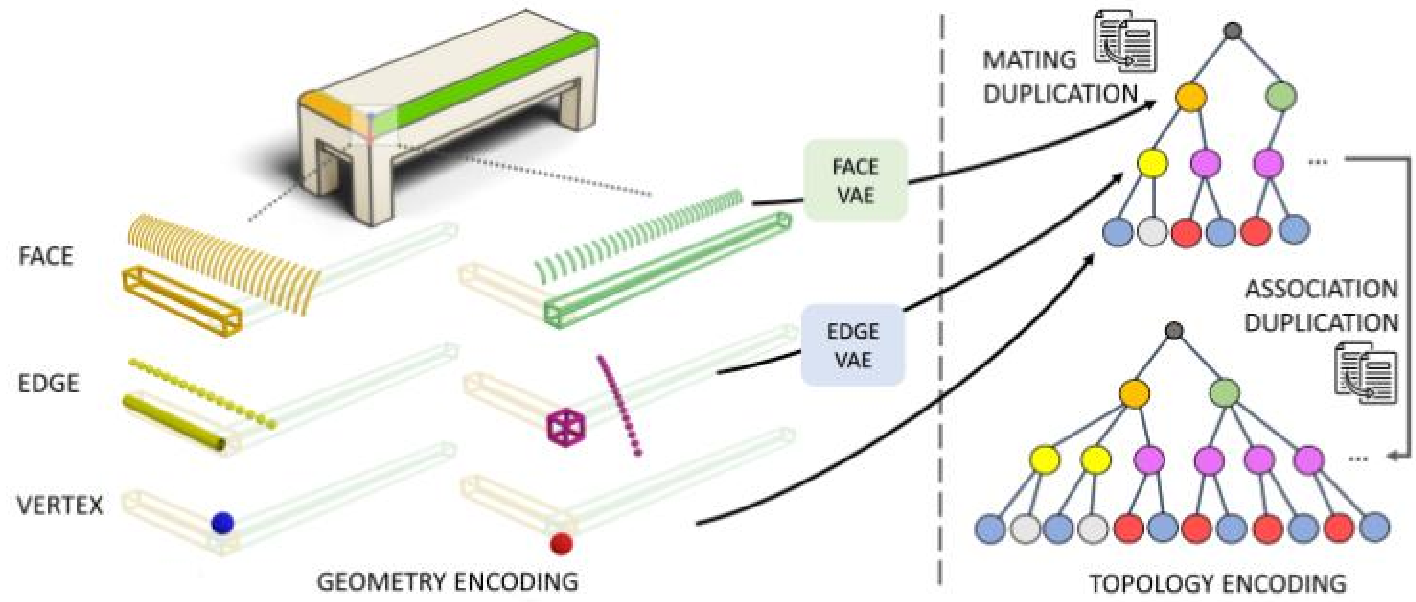


Topology Encoding by Node Duplication

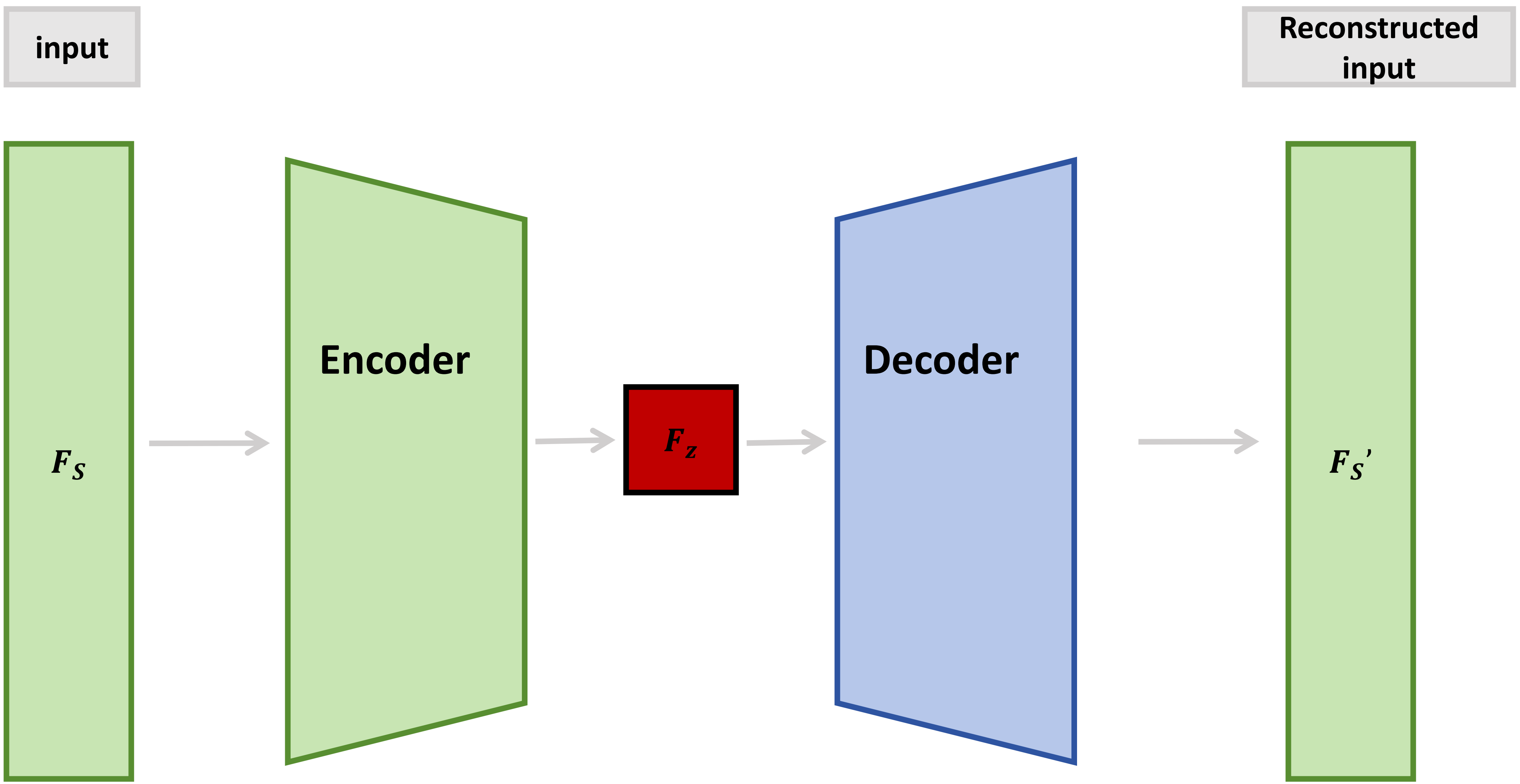
Association Duplication

The number of edges bounding a face and the faces forming a solid vary, which is the last ambiguity in the graph topology of our tree representation.

Our idea is to pick a predetermined maximum branching factor for each tree level and pad randomly selected nodes at each parent node.



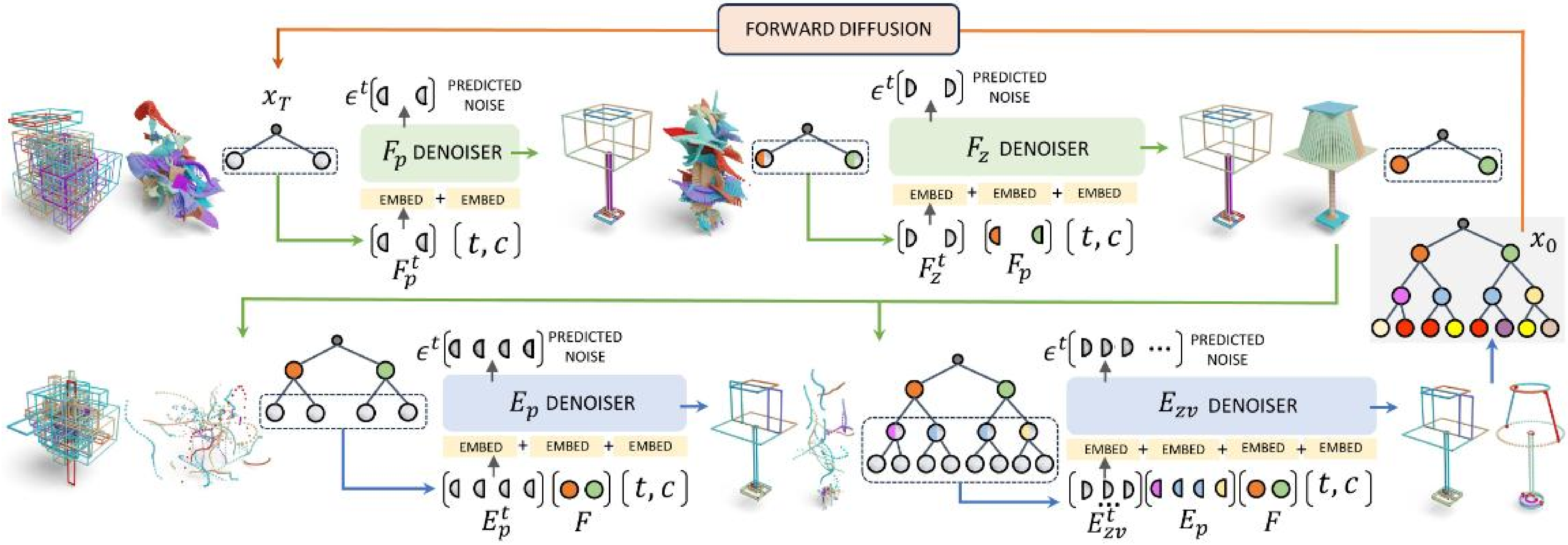
Shape Geometry VAE



Latent Diffusion Module

We follow the DDPM training scheme and use four Transformer-based denoisers to sequentially remove the noise added to the nodes.

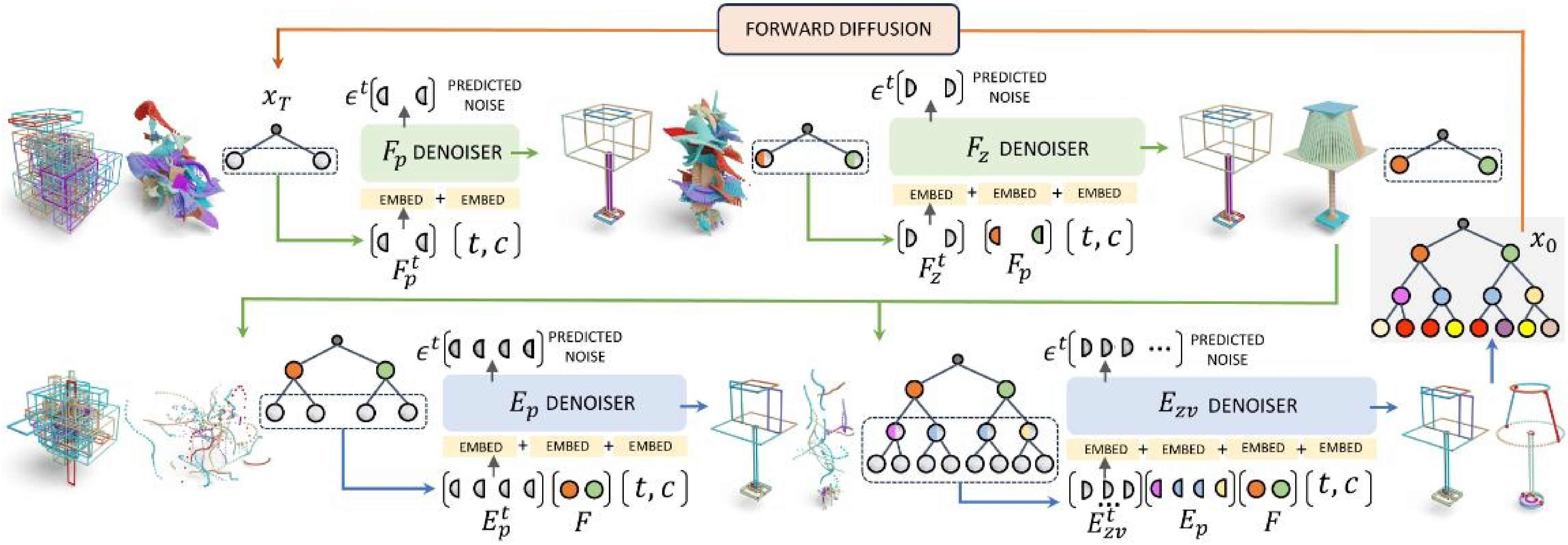
$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}),$$
$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t,$$



Latent Diffusion Module

Generating all geometry at once is difficult. We use sequential generation to denoise the face, edge, and vertex progressively

$$p(\mathbf{x}) = p(E_{zv}|E_p, F)p(E_p|F)p(F_z|F_p)p(F_p|\emptyset).$$



Latent Diffusion Module

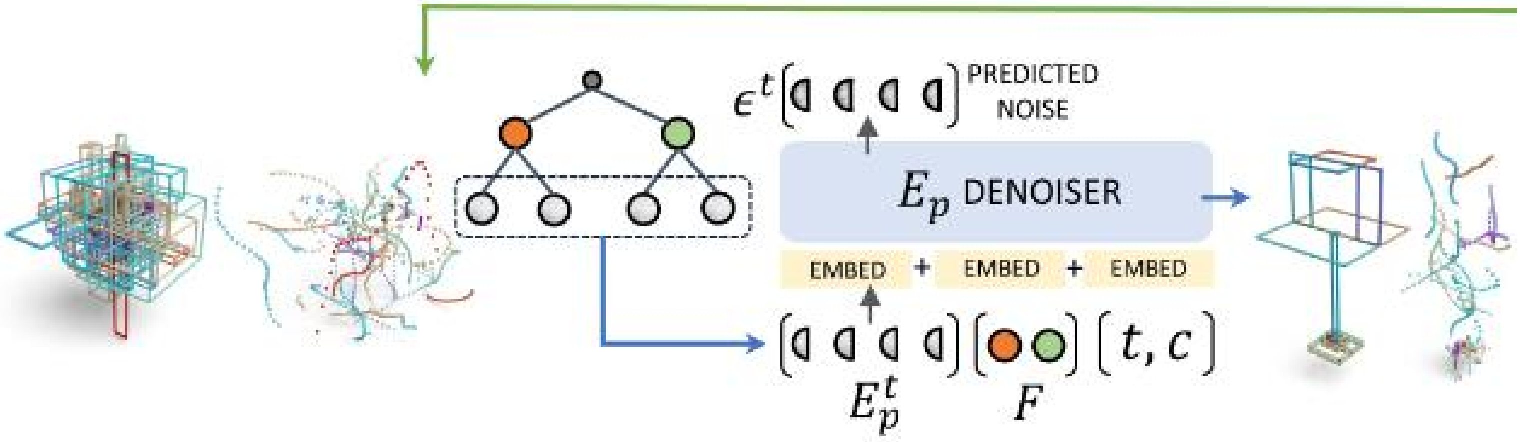
The four denoising networks in our latent diffusion module share a common Transformer backbone.

Without loss of generality, we use edge position denoiser as an example.
Network is conditioned on the previously denoised face tokens.

$$\boldsymbol{F} \leftarrow \mathrm{MLP}(W_p F_p) + \mathrm{MLP}(W_z F_z).$$

Rather than using cross-attention, the predefined parentchild relations from the tree are used to directly inject the face condition.

$$\hat{\boldsymbol{E}}_{\boldsymbol{p},\boldsymbol{j}} \leftarrow \boldsymbol{E}_{\boldsymbol{p},\boldsymbol{j}} + \boldsymbol{F}_{\boldsymbol{i}}.$$



Latent Diffusion Module

All four Transformer-based denoisers are trained separately to predict the L2-norm regression loss of the added sampled noise as in DDPM.

During sampling, the predicted noise at every time step is used to denoise the data from random Gaussian noise.

$$L = \mathbb{E}_{t,\mathbf{x}_0,\epsilon_t}\left[\|\epsilon_t - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t)\|^2\right],$$

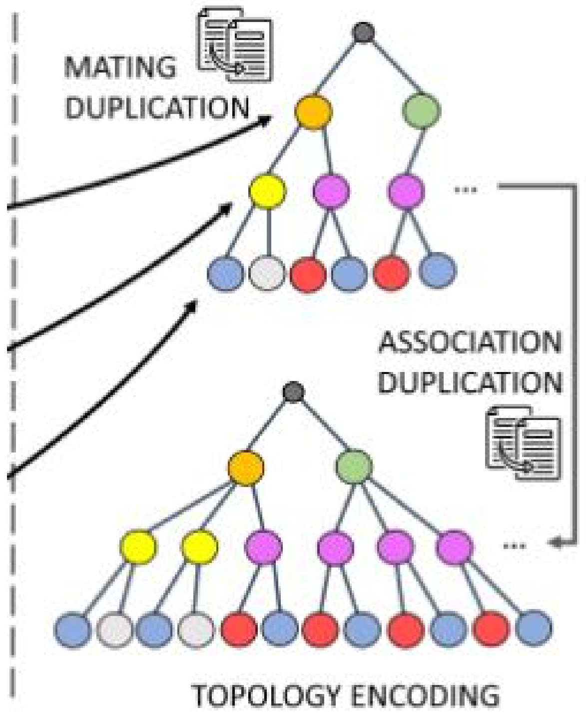
B-rep Post-Processing

Given the **face nodes**, we detect duplicates as those with bounding box corners within 0.08 Euclidean distance of each other and averaged pointwise difference less than 0.2 for the decoded points from the latent shape feature.

A similar procedure recovers the unique **edges** associated under every face.

The **vertex**/edge association is always two and does not require deduplication.

Finally, we traverse the tree again from leaf to root to find duplicated child nodes at the same level but associated with different parents, which are the shared edges and vertices for determining the final mating relations.



Experiments

Datasets

Generation is evaluated on the 1) DeepCAD dataset of mechanical parts made from sketch and extrude operations, and 2) the Furniture B-rep Dataset with more complicated furniture models.

B-reps with more than 30 faces or 20 edges per face, and made from multiple bodies are removed. After filtering, a total of 87,815 B-reps are used for training the VAEs and the latent diffusion module.

Experiments

Evaluation Metrics

For *Distribution Metrics* we use 3,000 Breps randomly-sampled from the generated data and 1,000 B-reps from the reference test set. For each B-rep, we sample 2,000 points from the solid surface and compute the following metrics:

Coverage (COV) is **the percentage of reference data with at least one match** after assigning every generated data to its closest neighbor in the reference set based on Chamfer Distance (CD).

Minimum Matching Distance (MMD) is **the averaged CD** between a reference set data and its nearest neighbor in the generated data.

Jensen-Shannon Divergence (JSD) measures **the distribution distance** between reference and generated data after converting point clouds into 28^3 discrete voxels.

Experiments

Evaluation Metrics

For *CAD Metrics* the same 3,000 B-reps are used to compute the following metrics

Novel percentage of data that do not appear in training set

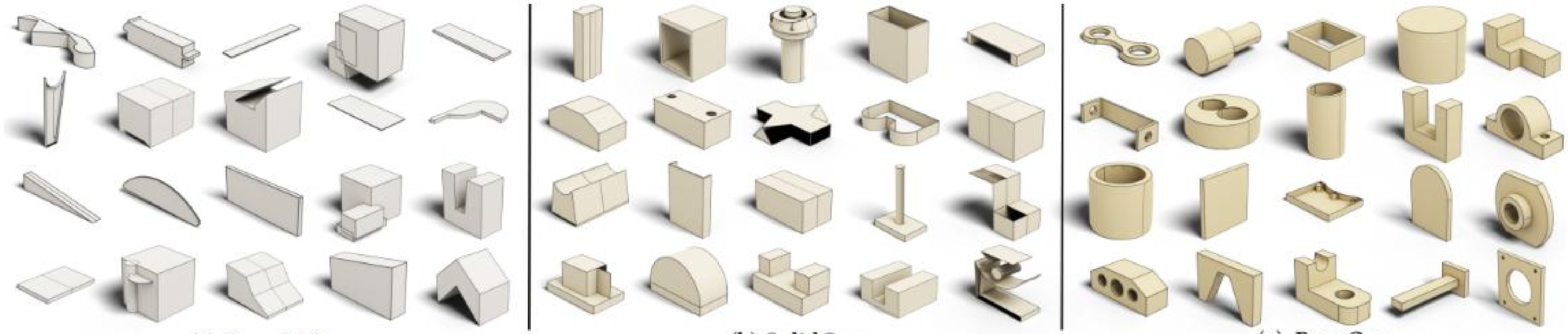
Unique percentage of data that appears only once in generation.

Valid percentage of data that are water-tight solids.

Unconditional B-rep Generation

BrepGen consistently outperforms both baselines with better COV, MMD scores and a substantially lower JSD, demonstrating improvements in generation quality and a closer match to the ground-truth distribution.

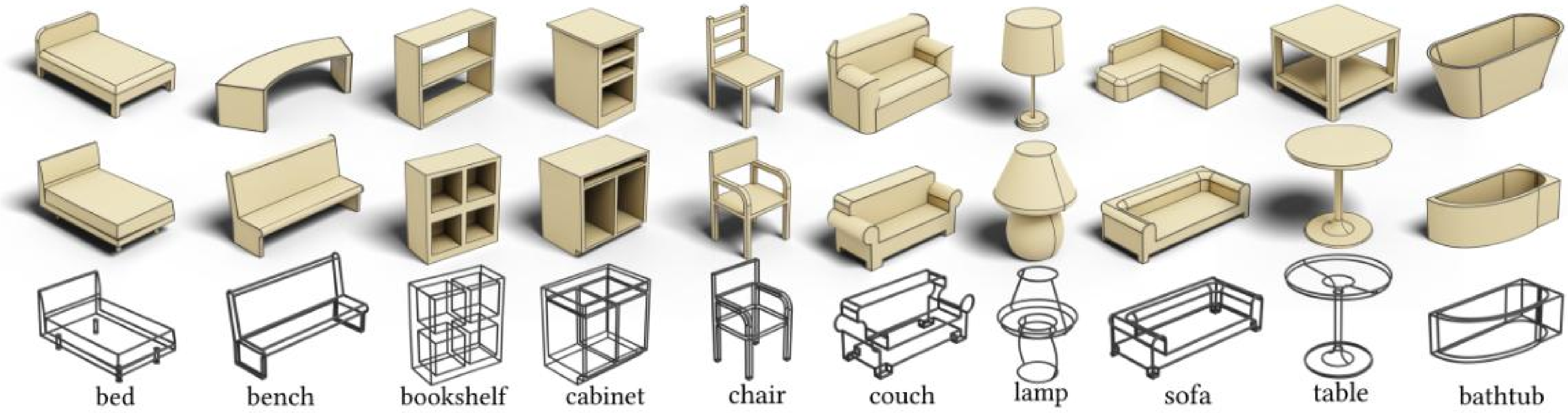
Method	COV % ↑	MMD ↓	JSD ↓	Novel % ↑	Unique % ↑	Valid % ↑
DeepCAD	65.46	1.29	1.67	87.4	89.3	46.1
SolidGen	71.03	1.08	1.31	99.1	96.2	60.3
BrepGen	71.26	1.04	0.09	99.8	99.7	62.9



(a) DeepCAD (b) SolidGen (c) *BrepGen*

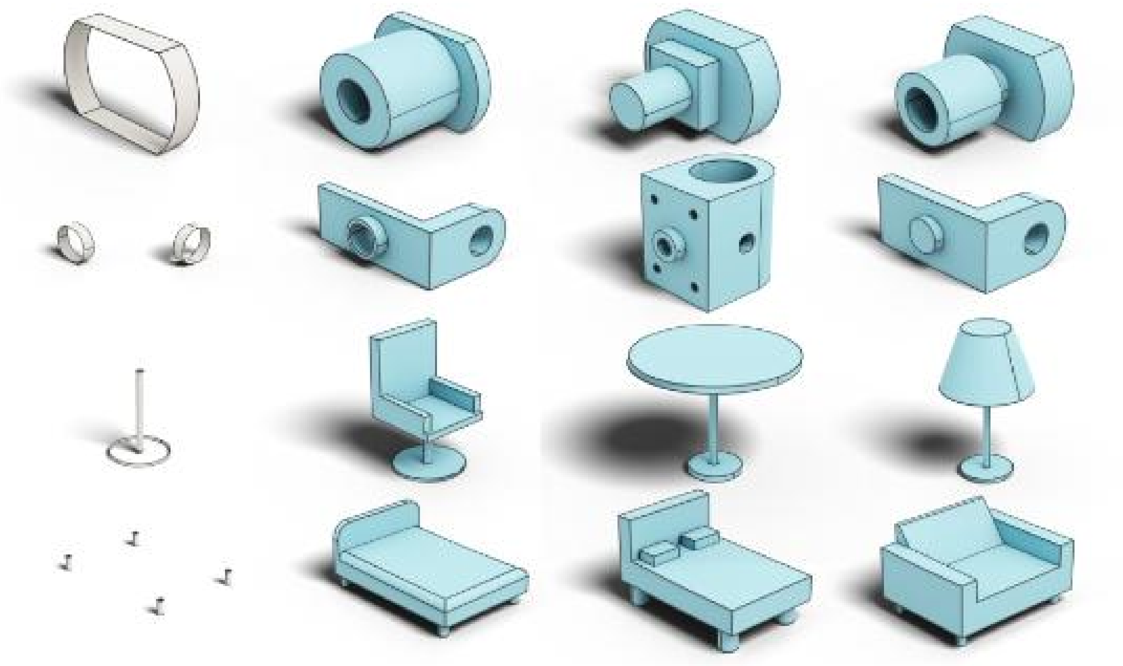
Controllable B-rep Generation

Conditional Generation.
A class-conditioned BrepGen is trained on the Furniture B-rep Dataset using classifierfree guidance. Class embeddings are added to every input token



Controllable B-rep Generation

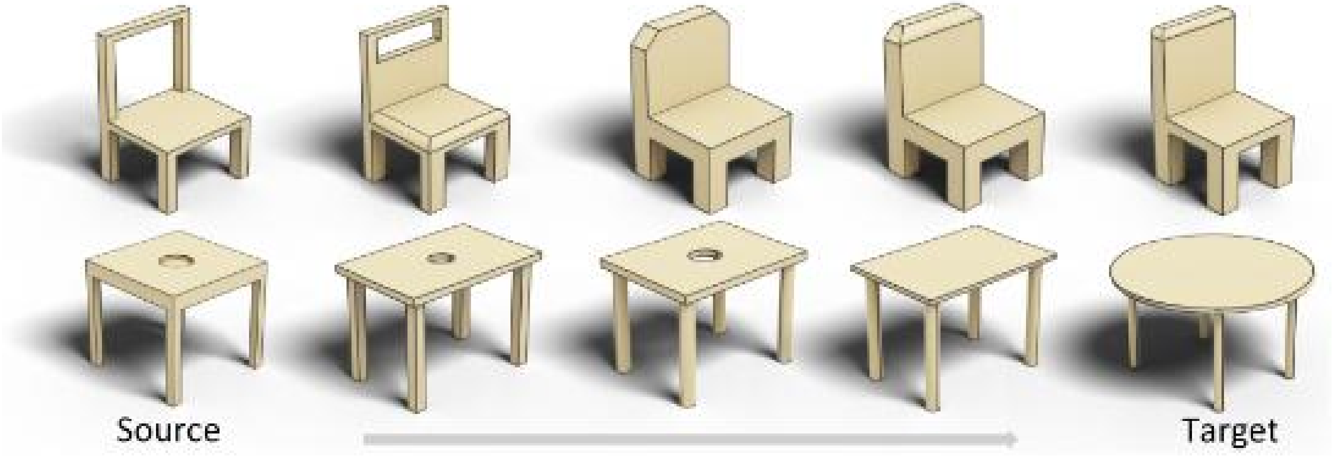
CAD Autocompletion.
We use a pretrained BrepGen to autocomplete the full B-rep from partial faces provided by the user.
A random subset of face tokens are replaced with the provided face geometry diffused to that time step during face denoising.



Controllable B-rep Generation

Design Interpolation.
Given a pair of CAD models, face tokens from the two shapes are concatenated, diffused for 150 steps, repeatedly padded and passed to the face denoiser.Edges and vertices are also regenerated.

From left to right, face tokens from the target are increasingly added to the source for face generation, after which the source tokens are removed until only the target tokens are left.



Limitations and Future Work

BrepGen supports only water-tight single body solids; more complicated CAD models with multiple assembled bodies are left to future work.

If edges or faces are too close to one another they will be merged and identified as one after deduplication.

Finally, while the heuristic post-processing used to generate the final B-rep is simple, fast, and can handle complicated data, to achieve better results future work on a learning-based post-processing module may provide more robust handling of invalid shapes