

BrepGiff: Lightweight Generation of Complex B-rep with 3D GAT Diffusion

CVPR 2025

Hao Guo · Xiaoshui Huang · Hao Jiacheng · Yunpeng Bai · Hongping Gan · Yilei Shi

목차

1. Introduction
2. Related Work
3. 3D Graph-Based Encoding
4. GAT-based Diffusion
5. Experiments
6. Conclusion

Abstract

- **기존 방법의 한계**
 - 기존 방법: 명령어 시퀀스 또는 Constructive Solid Geometry (CSG) 사용
 - 한계: 복잡한 디자인에 필요한 세밀한 기하학적 및 위상 정보를 완전히 포착하지 못함
- **B-rep (Boundary Representation)의 장점**
 - 3D 모델을 면, 모서리, 꼭짓점으로 표현하여 기하학적 및 위상 정보를 포괄적으로 제공
 - 하지만 복잡한 모델 생성 시 추출 및 통합의 어려움 존재
- **BrepGiff 소개**
 - 기술: 3D 그래프를 활용한 경량화된 접근법
 - 그래프 표현: 각 노드는 B-rep 모델의 면을, 엣지는 면 간 위상 관계를 나타냄
 - 최적화: Minimum Spanning Tree (MST)를 사용하여 계산 효율성 향상
 - 네트워크: Degree-Guided 3D Graph Attention Network (GAT) 디퓨전 모델 사용
- **생성 과정**
 - 디퓨전 모델: 엣지와 노드에 노이즈 추가 후 GAT 모델을 통한 예측 및 복원
 - 효율성: 활성 노드 예측을 통한 엣지 생성으로 처리 노드 수 감소
 - 제약 조건: 노드 도수 가이드와 위상 제약 조건을 통한 정확한 엣지 생성
- **결과 및 기여**
 - 위상 및 기하학적 특징 통합된 경량화된 3D 그래프 기반 B-rep 표현
 - 위상 제약 조건을 가진 GAT 확산 모델을 통한 효율적이고 고품질 생성
 - 복잡한 B-rep 모델 (면 수 100개 이상) 생성에서 최첨단 성능 달성
 - 적은 GPU 자원(2개의 RTX 4090)으로 훈련 가능

1. Introduction - 연구 배경

- CAD 모델은 거의 모든 산업 제품의 설계에 필수적이지만 **노동집약적이며 복잡한 작업임**
- 기존 자동 CAD 생성 기법
 - Command sequence 기반
 - CSG(Constructive Solid Geometry) 기반
 - 모두 **세밀한 geometry/topology 표현 한계**
- B-rep은 면·모서리·곡면의 **정확한 기하 + 위상 구조 모두 표현**
 - CAD의 가장 정확한 표현 방식
- 그러나 B-rep **직접 생성**은 어려움:
 - 파라메트릭 구조 처리 난이도
 - geometry & topology 통합 인코딩 어려움
- 이전 SOTA인 **BrepGen**은 50 faces 제한을 넘지 못함

1. Introduction — 논문의 핵심 아이디어

BrepGiff는 다음을 결합하여 **100+ face** 복잡 B-rep 생성을 달성:

1. B-rep → 3D Graph로 변환
 - 면(Face) → node
 - 모서리(Edge) → graph edge
 - UV 도메인 기반 centroid & sampling 활용
2. Topology + Geometry를 함께 표현하는 feature 설계
3. Degree-guided Graph Diffusion 모델 도입
 - Edge 삭제 기반 noising
 - Active node 중심 local-to-global 복원
4. GAT 기반 denoising
 - Node-edge 간 cross-attention으로 joint 복원
5. OpenCascade 기반 B-rep 재구성 pipeline

2. Related Work — CAD Generation

기존 CAD 생성 방식 분류

- Sketch → Extrusion 기반
 - ExtrudeNet, SECAD-Net 등
 - 명령어 sequence 생성
 - 그러나 최종 **Geometry** 정보 부족, Topology 제약 반영 어려움
- CSG(Constructive Solid Geometry) 기반
 - 모두 **세밀한 Geometry/Topology** 표현 한계
 - 기본 primitive들의 Boolean 조합
 - 해석이 용이하지만 **산업용 CAD에는 부적합**
(복잡한 곡면·freeform 표현에 한계)
- Point cloud 기반 CAD 복원
 - Point2Cyl, MV2Cyl, Split-and-Fit 등
 - 간접적 reconstruction → topology fidelity 낮음

대부분 *geometry/topology*가 분리되어 있으며, 정밀한 *B-rep* 구조를 직접 생성하는 연구는 매우 적음

2. Related Work — B-rep Generation

기존 연구 특징

- B-rep segmentation / classification: UVNet, BRepNet 등
- B-rep 생성:
 - Point cloud segmentation → B-rep fitting 방식 (ComplexGen, Point2CAD 등)
 - 최근 B-rep을 직접 생성하는 시도:
 - SolidGen (Transformer 기반 autoregressive)
 - BrepGen (Latent diffusion 기반)
- 그러나 여전히 한계점이 있음:
 - Geometry & topology 통합 표현이 미성숙
 - Face 수가 증가하면 **GPU 자원 요구 증가**
 - BrepGen조차 50 faces 이상 처리 어려움

2. Related Work — 3D Diffusion Models

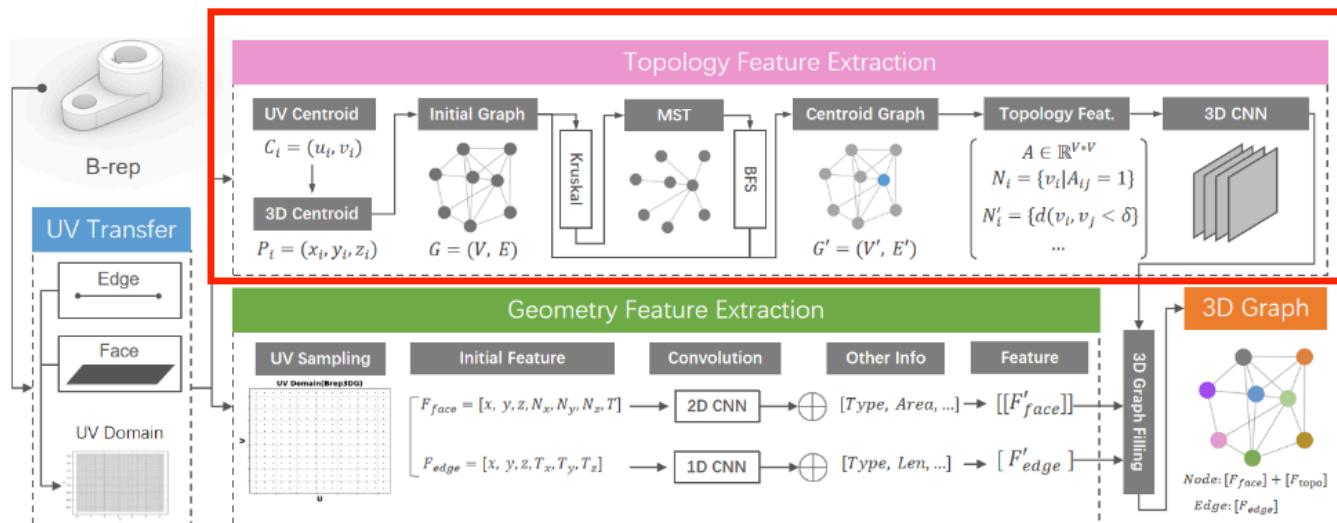
3D Diffusion의 활용

- 활용 사례
 - Mesh, SDF, voxel, point cloud 생성
 - Latent set diffusion(3DShape2VecSet)
 - MeshDiffusion, GetMesh 등
- 하지만 대부분 **mesh, point cloud** 위주로 B-rep의 topological constraints를 직접 다룬 모델은 없음

3. 3D Graph-Based Encoding — 개요

- B-rep 핵심 구성:
 - Topology: face adjacency, connectivity
 - Geometry: surfaces, normals 등
- 해결해야 하는 문제:
 - Topology 및 Geometry 특징을 각각 추출하는 방법
 - 위의 두 가지 특징을 통합하는 방법
- BrepGiff에서는 다음과 같이 해결:
 - B-rep 모델을 **3D Graph**로 인코딩
 - Face centroid = Node
 - Adjacency = Edges
 - MST로 복잡성을 줄이고 Centered Graph 구성
 - Topology & Geometry를 각각 추출해 **Node Feature / Edge Feature**로 통합

3.1 Topology Encoding



Graph Construction

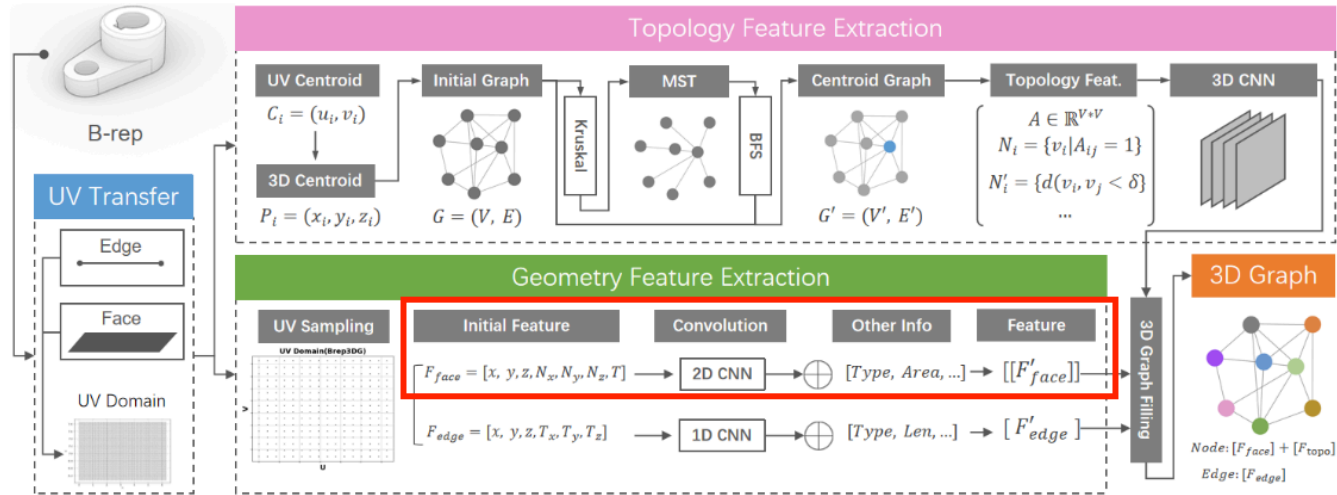
1. Face Centroid 계산: UV Domain에서 각 Face의 Centroid 계산
2. 3D 좌표 획득: Face Centroid CAD 좌표계로 매핑하여 실제 중심점 좌표(P_i) 획득
3. 초기 그래프 생성: 모든 Face Centroid 3D 좌표를 Node로 하는 3D 그래프 생성
4. MST 구축: Kruskal MST로 Edge 수 최소화 (사이클 제거)
5. Centroid Graph 구축: BFS로 연결성이 가장 높은 면을 찾아 그래프의 root로 사용

3.1 Topology Encoding

Topology Feature 구성 요소

- 다음 정보를 추출하여 벡터 $X = [A, N, C, D]$ 를 형성
 - A : Adjacency Matrix → 그래프 내 노드 간의 연결 관계를 나타내는 행렬 / 누구와 붙어 있는가?
 - $A \in \mathbb{R}^{V \times V}$ (V :노드 개수)
 - $A_{ij} = 1$: 정점 v_i 와 v_j 사이에 간선(edge) e_{ij} 이 존재하면 1, 존재하지 않으면 0
 - N : Direct + Potential Neighbors → 모든 노드에 대한 이웃 정보를 모아 놓은 행렬 / 누가 가까이 있는가?
 - $N = [n_1, n_2, \dots, n_V]^T, N \in \mathbb{R}^{V \times k}$ (k : dimension of neighbor information)
 - Direct Neighbors ($N_i = \{v_j \mid A_{ij} = 1\}$): v_i 와 직접 연결된 노드들의 집합
 - Potential Neighbors ($N'_i = \{v_j \mid d(v_i, v_j) < \delta\}$): 직접 연결되어 있지는 않지만 정점 v_i 와 v_j 사이의 거리 $d(v_i, v_j)$ 가 미리 정의된 거리 임계값 δ 보다 작은 노드들의 집합
 - C : Euler Characteristic, Loop Count → 두 가지 전역적인 위상 속성을 결합한 2차원 벡터 / 모델의 전체적인 위상 구조는?
 - $C = [L, \chi]^T, C \in \mathbb{R}^2$
 - L : 그래프 내의 루프(cycles) 수, 모델의 구멍(holes)이나 닫힌 경로의 수를 나타낼 수 있음
 - χ : 그래프의 기본 형태를 나타내는 값으로, $\chi = V - E + F$ 계산
 - 여기서 V 는 CAD vertex의 수, E 간선의 수, F 면(여기서는 원래 B-rep의 면에 해당함)
 - D : Degree Distribution → 각 노드의 차수 정보를 담고 있는 행렬 / 노드의 복잡도는?
 - $D = [d_1, d_2, \dots, d_V]^T, D \in \mathbb{R}^V$
 - d_i : 정점 v_i 에 연결된 엣지의 총 개수
- 모든 topology feature는 3D CNN을 통해 32-dim embedding으로 축약됨

3.2 Geometry Encoding

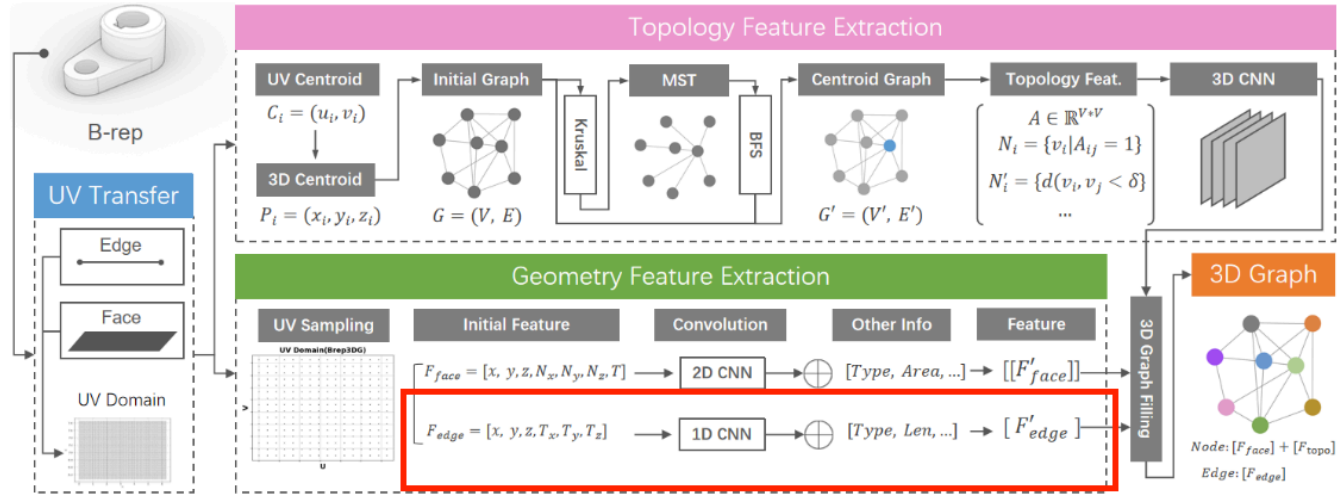


Face Geometry

- Parametric Surface를 UV 도메인으로 변환 후, UV 도메인에서 균일하게 32×32 포인트 샘플링
- 각 샘플링 포인트는 아래 정보를 포함하는 7차원 벡터로 구성
 - 3D 좌표 $[X, Y, Z]$
 - Surface Normal: $[N_x, N_y, N_z]$
 - Trimming Surface와의 관계: T
- 샘플링된 포인트 feature($f_{geo} \in \mathbb{R}^{32 \times 32 \times 7}$)에 2D CNN 컨벌루션을 적용하여 차원을 축소하고 128차원 벡터 F_{geo} 를 얻음
- F_{geo} 외에 면의 4가지 속성 feature를 계산 (각각 32차원 벡터)
 - 유형: F_{typ} (평면, 원기둥, 구면, ..)
 - 면적: F_{area}
 - 차수(degree): F_{deg} (해당 face가 몇 개의 다른 face와 인접하는지)
 - 인덱스: F_{indx} (face의 정렬된 index)
- 이 5가지 특징을 모두 연결(concatenate)하여 최종 면 특징 벡터 생성

$$F'_{face} = \text{Concat}(F_{geo}, F_{typ}, F_{area}, F_{deg}, F_{indx}) \in \mathbb{R}^{256}$$

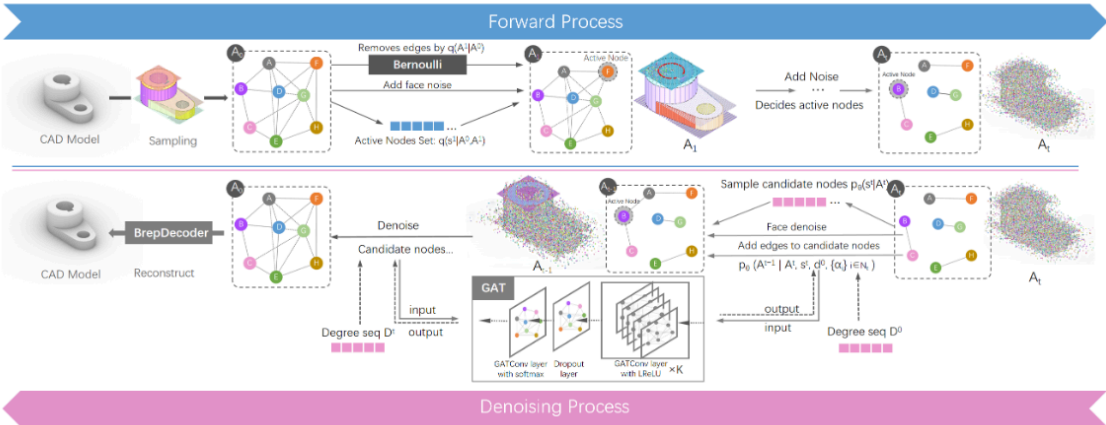
3.2 Geometry Encoding



Edge Geometry

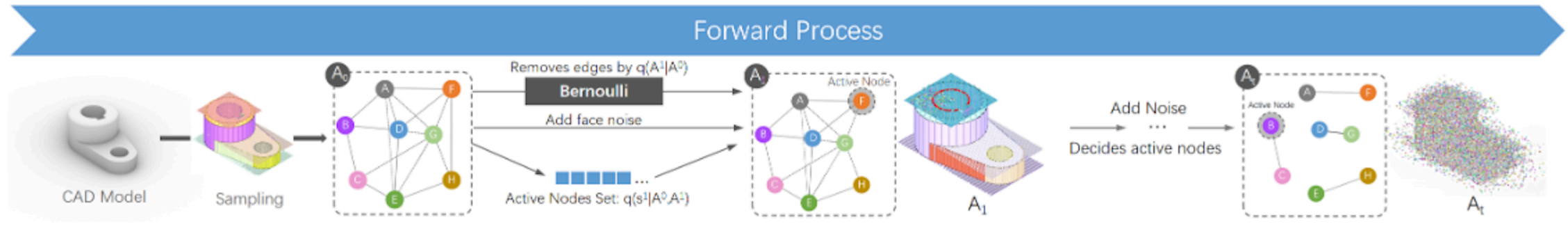
- UV 도메인에서 포인트를 균일하게 샘플링
- 각 샘플링 포인트는 아래 정보를 포함하는 6차원 벡터로 구성
 - 3D 좌표 $[X, Y, Z]$
 - 접선 벡터(Tangent Vectors): $[T_x, T_y, T_z]$
- 샘플링된 포인트 feature($e_{geo} \in \mathbb{R}^{m \times 6}$)에 1D CNN 컨볼루션을 적용하여 차원을 축소하고 128차원 벡터 E_{geo} 를 얻음
- E_{geo} 외에 Edge의 4가지 속성 feature를 계산 (각각 32차원 벡터)
 - 유형: E_{typ} (직선, 원호, 타원, ..)
 - 길이: E_{len}
 - 블록성(convexity): E_{con} (블록/오목)
 - 인접 면들의 차수 시퀀스: E_{adj} (해당 edge에 붙어있는 두 face의 degree)
- 이 5가지 특징을 모두 연결(concatenate)하여 최종 Edge 특징 벡터 생성

$$F'_{edge} = \text{Concat}(E_{geo}, E_{typ}, E_{len}, E_{deg}, E_{indx}) \in \mathbb{R}^{256}$$



- 기존 DDPM을 그래프 구조에 맞게 수정

4.1 Forward Noise Process



- DDPM을 기반으로 B-rep 데이터 생성에 특화된 노이즈 추가 방식 설계 → **Edge**를 점점 없애고 **Feature**에 노이즈 추가
- **Topoogy(Edge) 노이즈 추가 방식**
 - 그래프의 Edge를 삭제하여 점점 연결성이 사라지는 방향으로 노이즈 추가
 - 삭제 여부는 Node의 차수(Degree)를 기준으로 확률적으로 결정
 - Degree가 낮은 노드는 Edge가 더 쉽게 지워지고 Degree가 높은 Edge는 잘 안지워짐 → Bernoulli 확률로 모델링
- **Geometry(Node/Edge) 노이즈 추가 방식**
 - Node Feature(Centroid Geometry Feature)와 Edge Feature(Edge Geometry Feature)에 Gaussian noise 를 스케줄에 따라 직접 더함
 - 일반 DDPM처럼 noise variance schedule β_t 사용

4.1 Forward Noise Process

- Forward transition 표현

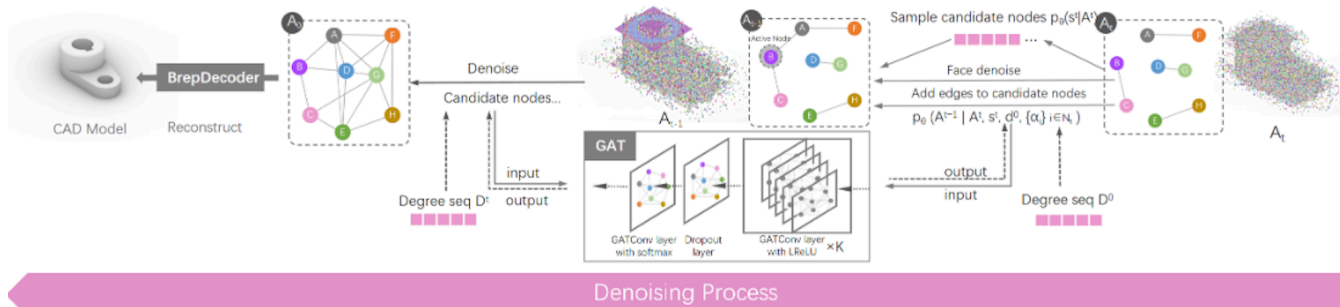
$$q(\mathbf{A}^{1:T}, \mathbf{s}^{1:T} | \mathbf{A}^0) = \prod_{t=1}^T q(\mathbf{A}^t | \mathbf{A}^{t-1}) q(\mathbf{s}^t | \mathbf{A}^{t-1}, \mathbf{A}^t)$$

- $\mathbf{A}_t$: 시간 t 에서의 Adjacency Matrix
- $\mathbf{s}_t$: 시간 t 에서의 Active Node 상태
- $q(\mathbf{A}^{1:T}, \mathbf{s}^{1:T} | \mathbf{A}^0)$: 초기 그래프 ($\mathbf{A}^0$)가 주어졌을 때, 전체 노이즈 시퀀스($\mathbf{A}^{1:T}$)와 활성 노드 상태 시퀀스 ($\mathbf{s}^{1:T}$)가 발생할 결합 확률(joint probability)
- $\prod_{t=1}^T$: $t = 1$ 부터 T 까지의 모든 항을 곱하는 연산자
- $q(\mathbf{A}^t | \mathbf{A}^{t-1})$: 이전 시간 단계의 인접 행렬($\mathbf{A}^{t-1}$)이 주어졌을 때, 현재 시간 단계의 인접 행렬 ($\mathbf{A}^t$)이 될 조건부 확률
- $q(\mathbf{s}^t | \mathbf{A}^{t-1}, \mathbf{A}^t)$: 이전 시간 단계 ($\mathbf{A}^{t-1}$)과 현재 시간 단계 ($\mathbf{A}^t$)의 인접 행렬이 주어졌을 때, 현재 시간 단계의 활성 노드 상태 ($\mathbf{s}^t$)가 될 조건부 확률

시간이 지남에 따라 그래프가 점점 망가지는 과정이 일어나는 동시에 그 파괴의 영향을 받은 노드가 어디인지를 표시하는 모든 확률적 경로를 모델링 (이전 그래프 $\mathbf{A}^{t-1}$ 을 기반으로 이번 그래프 $\mathbf{A}^t$ 가 될 확률)과 (그 과정에서 어떤 노드가 바뀌었는지 표시하는 $\mathbf{s}^t$ 가 될 확률)의 곱으로 표현

- Active Node Tracking
 - Edge가 변경된 Node만 별도로 복사하여 Active Node Set에 저장
 - Active node는 이후 역확산(Denoising) 과정에서 복원할 대상 설명 신호 역할
- Active Node 정의 공식
 - 노드의 차수가 바뀐 경우 Node는 Active 상태가 됨 (바뀌면 해당 노드는 1)
- EDGE 논문과의 차별점
 - 기존 EDGE는 Edge 제거가 반복되면 노드가 고립될 수 있음 (Degree=0)
 - BrepGiff는 각 노드의 degree가 최소 1이 되도록 edge 제거를 제한하여 고립 노드가 생기지 않게끔 함 (Graph Attention Network)는 최소 1개의 edge가 있어야 동작함)

4.2 GAT-based Denoising



- BrepGiff는 GAT(Graph Attention Network)를 사용하여 Topology + Geometry를 동시에 복원(Denoising)
- 기존 GAT는 노드 간 메시지 전달에 초점을 맞췄지만, BrepGiff는 노드 $\leftrightarrow$ 엣지 간 양방향 Cross-Attention 을 추가하여
 - Face feature (노드, 256-dim)
 - Edge feature (엣지, 256-dim)를 공동 최적화(Joint Optimization)

4.2 GAT-based Denoising — Cross Attention

- **Node-to-Edge Attention (Node→Edge)**

- 목적: 엣지 특징 업데이트 시, 연결된 노드들의 정보 반영
- 각 엣지 e_{ij} 에 대해 Attention Weight 계산

$$\alpha_{ij}^e = \text{softmax} \left(\text{LReLU} \left(\mathbf{a}_e^T [\mathbf{W}_n \mathbf{h}_i \| \mathbf{W}_n \mathbf{h}_j \| \mathbf{W}_e \mathbf{f}_{ij}] \right) \right)$$

- h_i, h_j : 노드 Feature (256-dim)
- f_{ij} : 엣지 Feature (256-dim)
- $\mathbf{W}_n, \mathbf{W}_e$: 노드/엣지 feature projection matrix (256×256)
- α_e : 엣지 업데이트 중요도를 계산하는 Attention 벡터 (756-dim)
- $\|$: concatenation

- **Edge-to-Node Attention (Edge→Node)**

- 목적: 노드 특징 업데이트 시, 연결된 엣지들의 정보 반영
- 각 노드 v_j 에 대해 Attention Weight 계산

$$\beta_i^n = \text{softmax} \left(\text{LReLU} \left(\mathbf{a}_n^T \left[\mathbf{W}_n h_i \| \mathbf{W}_e \text{AvgPool} \left(\{f_{ij}\}_{j \in N_i} \right) \right] \right) \right)$$

- N_i : 노드 v_i 와 연결된 이웃 엣지 집합
- α_n : 노드 업데이트 중요도를 계산하는 Attention 벡터 (512-dim)

4.2 GAT-based Denoising — Joint Feature Update

- BrepGiff는 Cross-Attention을 통해 계산된 **노드→엣지 attention** α^e , **엣지→노드 attention** β^n 을 이용해 엣지 및 노드 feature를 동시에 denoising

- **Edge Feature Denoising**

■ 면 i 와 면 j 의 특징을 참고해 그 사이의 Edge의 특징을 재조정하는 과정

$$f'_{ij} = \sigma \left(\alpha_{ij}^e \cdot W_e [W_n h_i || W_n h_j || f_{ij}] \right)$$

- σ : Sigmoid function
- α^e : 노드→엣지 Cross-Attention
- h_i, h_j : 노드(face) feature
- f_{ij} : 기존 엣지(edge) feature
- W_n, W_e : 학습 가능한 projection matrix

- **Node Feature Denoising**

■ 자기 자신과 주변 면들의 영향으로 Face의 특징을 재조정하는 과정

$$h'_i = \sigma \left(\beta_i^n \cdot \mathbf{W}_n h_i + \sum_{j \in N_i} \gamma_{ij} \mathbf{W}_n h_j \right)$$

- β_i^n : 엣지→노드 Cross-Attention
- N_i : 노드 (v_i)의 주변 노드 집합
- γ_{ij} : Neighbor Aggregation Weight (GAT 기반 Attention)

4.2 GAT-based Denoising — Joint Feature Update

- Degree-Guided Local-to-Global Denoising

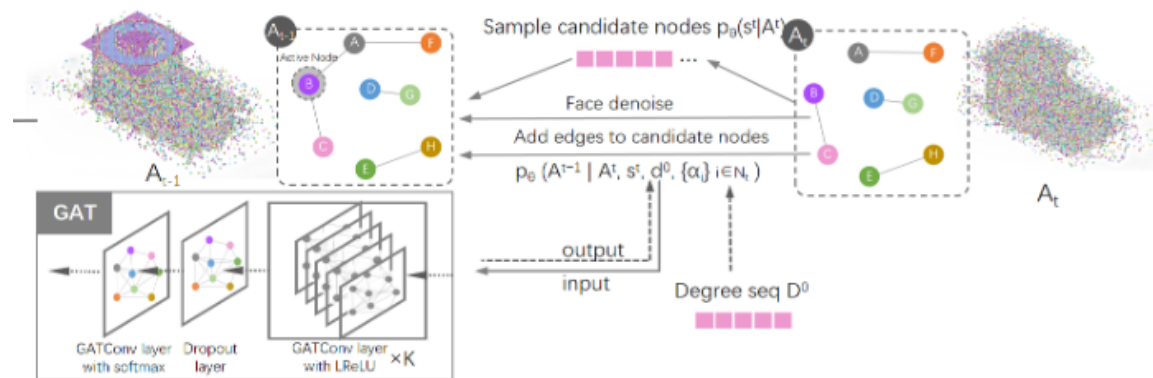
- Active Node는 **Forward** 과정에서 변화가 가장 컸던 노드이며 우선 복원해야 하는 대상
- 모델은 각 step에서
 - a. Active Node 탐지
 - b. 해당 노드에서 Edge 복원 확률 예측
 - c. Local → Global 순서로 점진적 그래프 재구성

- Reverse (Denoising) Transition 확률 모델

$$p_{\theta}(A_{0:T}, s_{1:T}, d_0 | A_T) = p_{\theta}(d_0)p_{\theta}(A_T) \times \prod_{t=1}^T \left[p_{\theta}(s_t | A_t, d_0) \prod_{i \in N_t} p_{\theta}(e_i | A_t, s_t, d_0, \alpha_i) \right]$$

- $p_{\theta}(d_0)$: 초기 Degree 분포
 - $p_{\theta}(A_T)$: 초기 Noisy Graph
 - $p_{\theta}(s_t | A_t, d_0)$: 현재 그래프(A_t)와 초기 노드 차수 d_0)가 주어졌을 때 time step t에서 어떤 노드들이 활성화될 것인지 s_t 를 예측하는 확률 분포
 - $p_{\theta}(e_i | A_t, s_t, d_0, \alpha_i)$: 현재 그래프(A_t), Activation Node 상태(s_t), 초기 노드 차수 d_0), GAT 를 통해 계산된 어텐션 가중치(α_i)가 주어졌을 때, Activation Node i 에 대한 엣지(e_i)의 존재 확률을 예측하는 분포
- 망가진 그래프를 원래 그래프로 되돌리기 위해 어떤 노드를 먼저 복원할지 정하고 그 노드에서 어떤 엣지를 복원해야 하는지 예측해 나가는 과정들을 모든 시간에 걸쳐 확률적으로 모델링 (복원 대상: Activation Node s_t , 엣지 존재 확률 e_i , 초기 Adjacency Matrix A_0)
 - Noisy Graph A_t 가 주어졌을 때 복원할 노드 s_t 예측, 그 노드의 엣지를 예측하여 A_{t-1} 예측, 이 과정을 반복

4.2 GAT-based Denoising — Joint Feature Update



Algorithm 1 GAT-Degree-based graph generation

Require: Empty graph A^T , graph model $p_\theta(A^{t-1}|A^t, s^t)$, degree sequence model $p_\theta(d^0)$, and diffusion steps T .

Ensure: Generated graph A^0

Draw $d^0 \sim p_\theta(d^0)$

for $t = T, \dots, 1$ **do**

Determine the active node set $\mathcal{N}_t$ based on the degree changes: $\mathcal{N}_t = \{i \mid s_i^t = 1\}$.

Compute attention weights α_i for each active node $i \in \mathcal{N}_t$ using GAT.

Draw $s^t \sim q(s^t \mid \deg(A^t), d^0)$.

Draw $A^{t-1} \sim p_\theta(A^{t-1} \mid A^t, s^t, d^0, \{\alpha_i\}_{i \in \mathcal{N}_t})$.

end for

원래 그래프의 degree 분포 d^0 를 예측하여 복원 순서를 결정하고, 각 스텝마다 Active Node를 선택한 뒤 GAT Attention을 이용해 해당 노드의 Edge를 복구함으로써 Noisy Graph A_T 를 원래 그래프 A^0 으로 점진적으로 되돌림

4.3 B-rep Reconstruction

- 디노이징된 3D 그래프를 유효한 B-rep 모델로 변환하기 위해 3단계 과정을 사용 (OpenCascade 사용)
 - i. Parametric Geometry Generation
 - Face → B-spline Surface 생성
 - 디노이징된 Face Point Cloud를 사용
 - `GeomAPI_PointsToBSplineSurface` 로 B-spline surface fitting
 - Edge → B-spline Curve 생성
 - 디노이징된 Edge Point Sequence를 기반
 - `GeomAPI_PointsToBSpline` 으로 curve parameterization
 - ii. Topology-Aware Trimming
 - GNN이 예측한 Adjacency Matrix를 기반으로 연결된 loop를 검출하여 surface boundary trim
 - Geometric 정합
 - Fitted surface/curve가 Node Feature에 맞도록
 - MSE loss로 Geometry alignment 수행
 - iii. Watertight Assembly
 - OpenCascade Topological Sewing을 사용해 Trimming 후 Faces들을 하나의 Manifold solid로 봉합
 - 학습 과정의 Loss 설계로 Self-Intersection 없이 CAD-valid 형식 자동 보장
- 해당 과정 결과 CAD 품질이 보장됨
 - B-spline fitting을 통해 기하학적 정확성이 보장됨
 - Reconstructed B-rep 모델은 산업용 CAD 유효성 기준을 충족함

5. Experiments — Dataset & Setup

Dataset

- ABC dataset (Geometric 딥러닝에서 널리 활용되는 대규모 CAD 모델 컬렉션)
- Unconditional Generation 평가를 위해 DeepCAD와 동일한 데이터셋 활용
- ABC 데이터셋 전처리 과정
 - NURBS 및 복잡한 표면을 가진 모델 제거
 - 최대 면 개수(faces) 제한 설정
 - 여러 형태를 가진 모델 및 중복 모델 삭제
- Max Face Count에 따라 여러 데이터 버전 구축

Training

- 2× RTX4090
- AdamW($5e-4$), Mixed Precision 사용
- 10,000 Epochs
- Batch size 64

5. Experiments — Evaluation Metrics

- **Coverage, COV**: 생성된 데이터 포인트가 참조 데이터 세트에서 가장 가까운 이웃과 매칭되는 참조 데이터의 비율
- **최소 매칭 거리(Minimum Matching Distance, MMD)**: 참조 데이터 포인트와 그들의 생성 데이터 세트 내 가장 가까운 이웃 사이의 평균 Chamfer 거리 (CD)
- **Jensen-Shannon Divergence, JSD**: 두 확률 분포 간의 대칭적인 Kullback-Leibler 발산 측정치
- **Novel**: 훈련 데이터 세트에 존재하지 않는 생성 모델의 비율 (B-rep 구조와 매개변수 특징 해싱을 통해 검증)
- **Unique**: 생성된 모델 중 중복되지 않은 모델의 비율 (정확한 해시 기반 중복 제거를 통해 보장)
- **Valid**: 오류 없이 물림새가 맞는 모델의 비율 (CAD 커널을 통한 위상 및 기하학적 검증으로 확인)

5. Experiments — Unconditional Generation

Table 1. Unconditional generation evaluation results comparison.

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	73.87	1.04	1.28	99.8	99.7	62.9
Ours	75.52	0.98	1.17	98.9	98.6	67.7

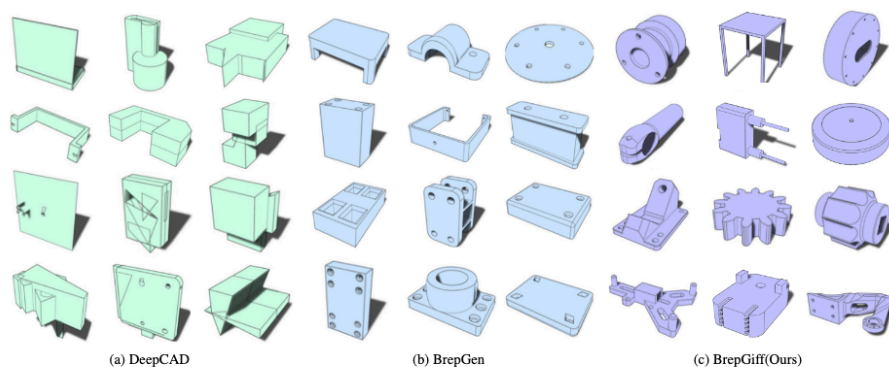


Figure 4. This figure demonstrates the generation result of: (a) DeepCAD, (b) BrepGen, and (c) BrepGiff(Ours). We sample CAD model of different complexity. BrepGiff is capable of generating high quality B-rep CAD models compared to other two techniques.

Table 2. Parameter quantity and training resource consumption.

Model	Parameter	Training Time	GPU	Max Faces
SolidGen	<i>unknown</i>	3days	A100×8	50
BrepGen	4.56×10^8	7days	RTX4090×2	50
Ours	7.80×10^7	4days	RTX4090×1	50

→ 베이스라인 대비 전반적인 성능 우수 및 설계 효율성

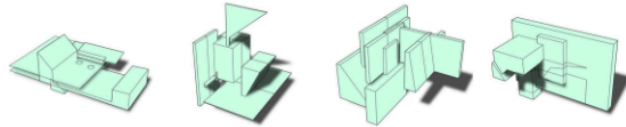
5. Experiments — Unconditional Generation

- Face에 따른 BrepGiff 성능

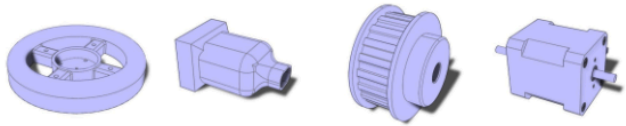
Table 3. Generation results for models with different complexities.

Max Face	COV %↑	MMD %↓	JSD %↓	Novel %↑	Unique %↑	Valid %↑
50	75.52	0.98	1.17	98.9	98.6	67.7
70	72.25	1.67	2.03	98.8	95.9	65.5
100	70.08	3.11	3.80	92.5	90.8	62.9
120	67.31	3.86	4.68	89.5	90.0	48.3

- 복잡한 모델에 대한 생성 성능



(a) DeepCAD



(b) BrepGiff

Figure 5. Generated complex CAD models (around 100 faces) comparison for models trained with 2 RTX4090s. Models generated by BrepGiff better represent the shape of real objects.

→ BrepGiff 생성 모델이 베이스라인 대비 더 현실적이고 정확한 형태 표현

5. Experiments — Failure Generation Cases

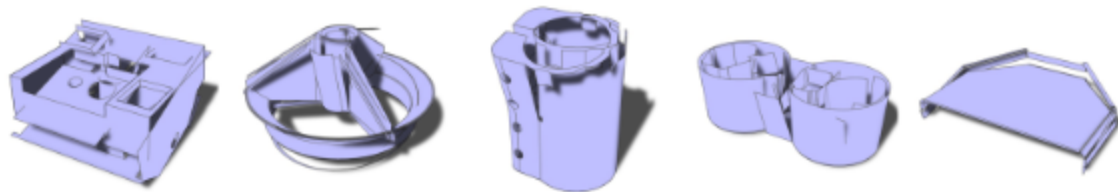


Figure 6. Failure cases generated by BrepGiff.

- 주요 문제점
 - Missing Faces: Face가 누락되는 현상 발생
 - Broken Faces: Face가 과도하게 처리되어 깨지는 현상 발생
- Topology는 잘 복원되나 Face 노드의 경우 **노이즈 레벨, 디퓨전 스텝의 적절성**에 따라 실패하게 됨
 - 불충분한 디퓨전 스텝: 노이즈 제거 과정이 너무 일찍 끝나면, 면이 누락되거나 불완전한 형태로 나타날 수 있음
 - 과도한 디퓨전 스텝: 노이즈 제거 과정이 너무 길면, 모델이 면의 기하학적인 정보를 과도하게 처리하여 원래 형태가 손상되거나 부서지는 결과가 발생할 수 있음
- 주로 복잡한 모델에서 해당 문제 발생

5. Experiments — Ablation studies

- 세 가지 핵심 구성 요소(CNN 기반 feature extraction, GAT 기반 denoising, diffusion step 수)의 효과를 평가
- Convolution Effect

Table 4. Ablation study on 1D and 2D CNN feature extractors.

Method	COV	MMD	JSD	Novel	Unique	Valid
	%↑	%↓	%↓	%↑	%↑	%↑
w/o 1D CNNs	55.22	8.50	12.40	77.5	78.0	50.3
w/o 2D CNNs	32.36	23.96	23.56	62.1	55.9	22.4
Full model	68.55	2.12	1.17	93.2	94.1	58.8

→ CNN이 Geometry 표현 학습에 필수적임을 입증

5. Experiments — Ablation studies

- GAT vs Non-GAT

Table 5. Ablation study on GNN denoising.

Method	COV %↑	MMD %↓	JSD %↓	Novel %↑	Unique %↑	Valid %↑
w/o GAT	45.52	18.55	20.03	65.6	69.4	22.9
w/ GAT	68.55	2.12	1.17	93.2	94.1	58.8

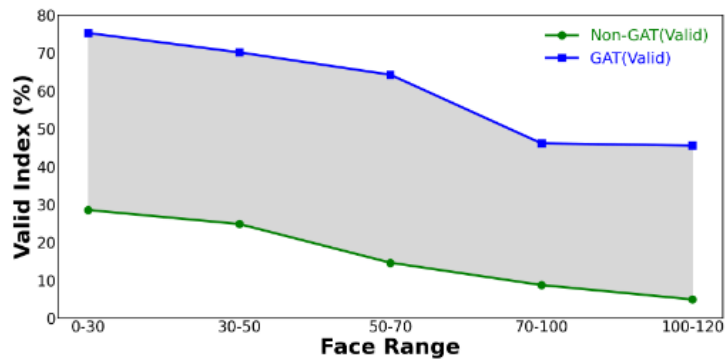


Figure 7. Valid index of GAT and non GAT generated data in different complexity intervals

→ GAT가 없으면 Edge-Face 정렬이 깨져 B-rep 연결성 불안정

5. Experiments — Ablation studies

- Diffusion Step

Table 6. Effect of diffusion steps on B-rep generation quality.

Step	COV %↑	MMD %↓	JSD %↓	Novel %↑	Unique %↑	Valid %↑
100	35.22	45.5	38.90	48.5	30.9	32.1
400	62.15	8.85	11.38	85.2	80.9	52.8
700	68.45	2.89	2.77	92.1	93.0	58.0
1000	68.55	2.12	1.17	93.2	94.1	58.8

→ 충분한 Diffusion Step 수가 고품질 B-rep 생성을 위해 필요함

6. Conclusion

- BrepGiff 소개: 새로운 경량 방법으로 복잡한 B-rep 모델을 고품질로 직접 생성
- 기술 핵심
 - 3D Graph Embedding 기법을 활용하여 위상적 특징과 기하학적 특징을 함께 인코딩
 - Diffusion Model을 통해 생성 과정에서 노이즈 추가
 - Degree-Guided Graph Attention Network (DGAT)를 사용하여 노이즈 제거
- 성능
 - 실험 결과, BrepGiff는 Brep 생성 작업에서 SOTA 성능 달성
 - 특히 복잡한 B-rep 모델 생성에서 뛰어난 성능 보임
- 해당 모델은 직접적인 B-rep 생성 기술 분야의 연구 및 응용에 기반 제공할 수 있을 것이라 기대됨

 **End of Review**
