

<https://arxiv.org/pdf/2105.09492>

# DeepCAD

## A Deep Generative Network for Computer-Aided Design Models

- | 김민성
- | 000-0000-0000
- | minsung1066@gmail.com

# 초록



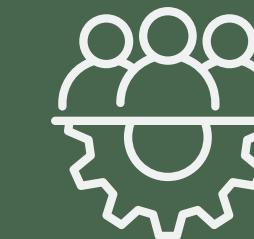
## 기존 연구

복셀(voxels), point clouds, polygon mesh 와 같은 방법을 통해 불연속적인 3D 형상 를 만드는 방법을 제시합니다.



## 기존 문제

CAD 작업의 불규칙적인 구조는 3D 생성 모델에 어려움이 있습니다.



## 논문 해결법 제안

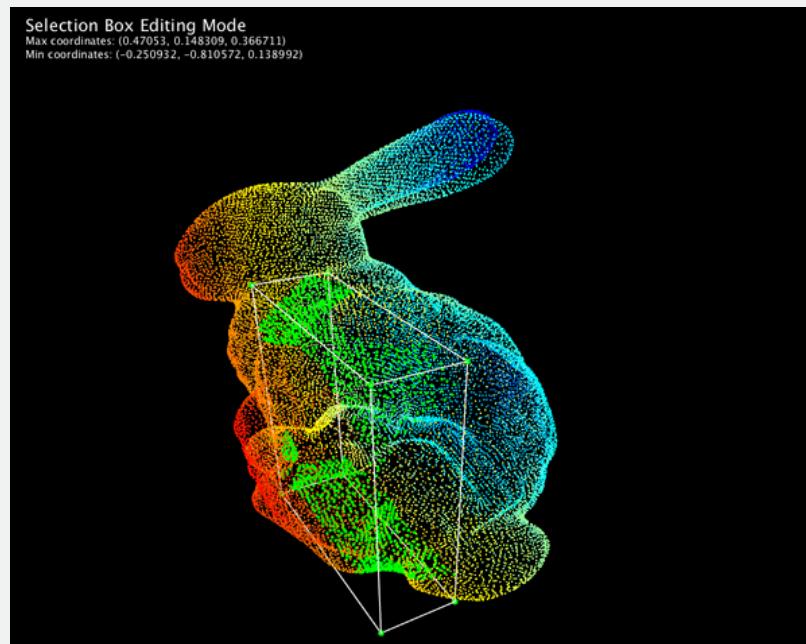
CAD를 통해 완전히 다른 방법으로 3D 생성 모델을 제시합니다.

CAD 작업물과 자연어 처리를 통해 모델의 형태를 유추하기 위한, Transformer 기반 CAD 생성 네트워크 모델을 제안합니다.

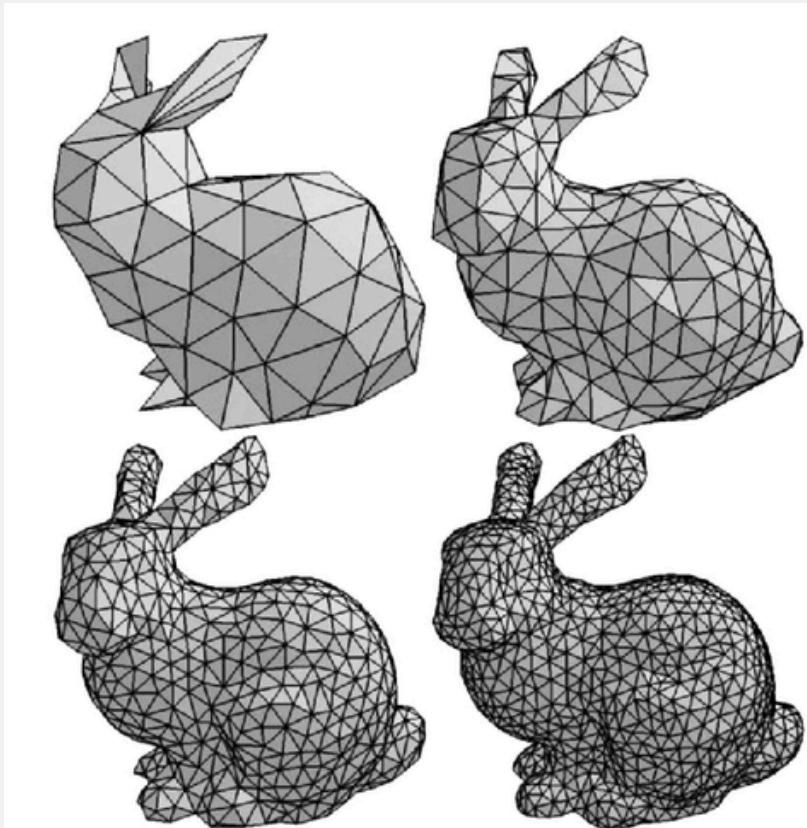
# 서론

Representations for 3D Shape Learning(3D Shape를 학습하기 위해 데이터 표현 방법)

3D Point Cloud



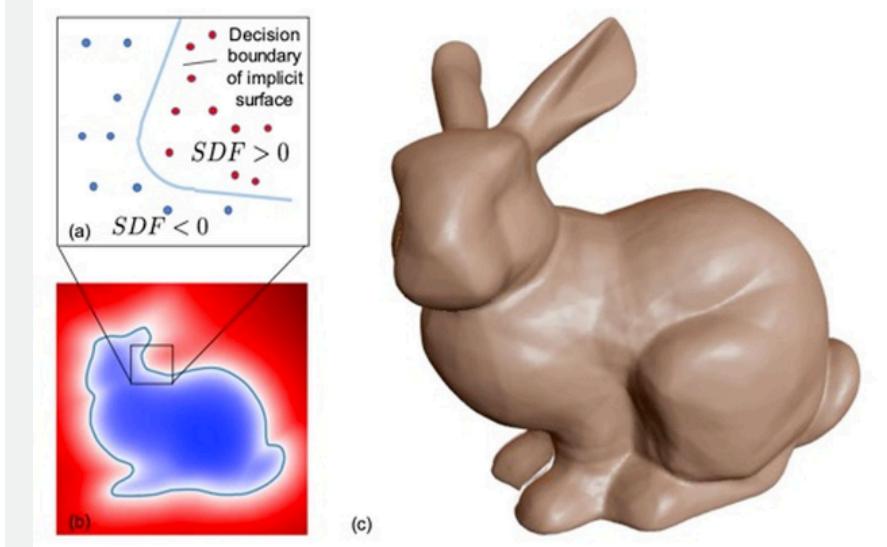
Polygon Mesh



Voxel



Occupancy  
해당 사진은 DeepSDF



# 서론

## 기존 연구

### 3D Point Cloud

- [6] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 40–49, Stockholm, Sweden, 10–15 Jul 2018. PMLR. [1](#), [2](#), [6](#), [7](#), [8](#), [4](#)
- [52] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4541–4550, 2019. [1](#), [2](#)
- [53] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation.
- [8] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. [1](#), [2](#)
- [30] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. 2019. [1](#), [2](#)

### Polygon Mesh

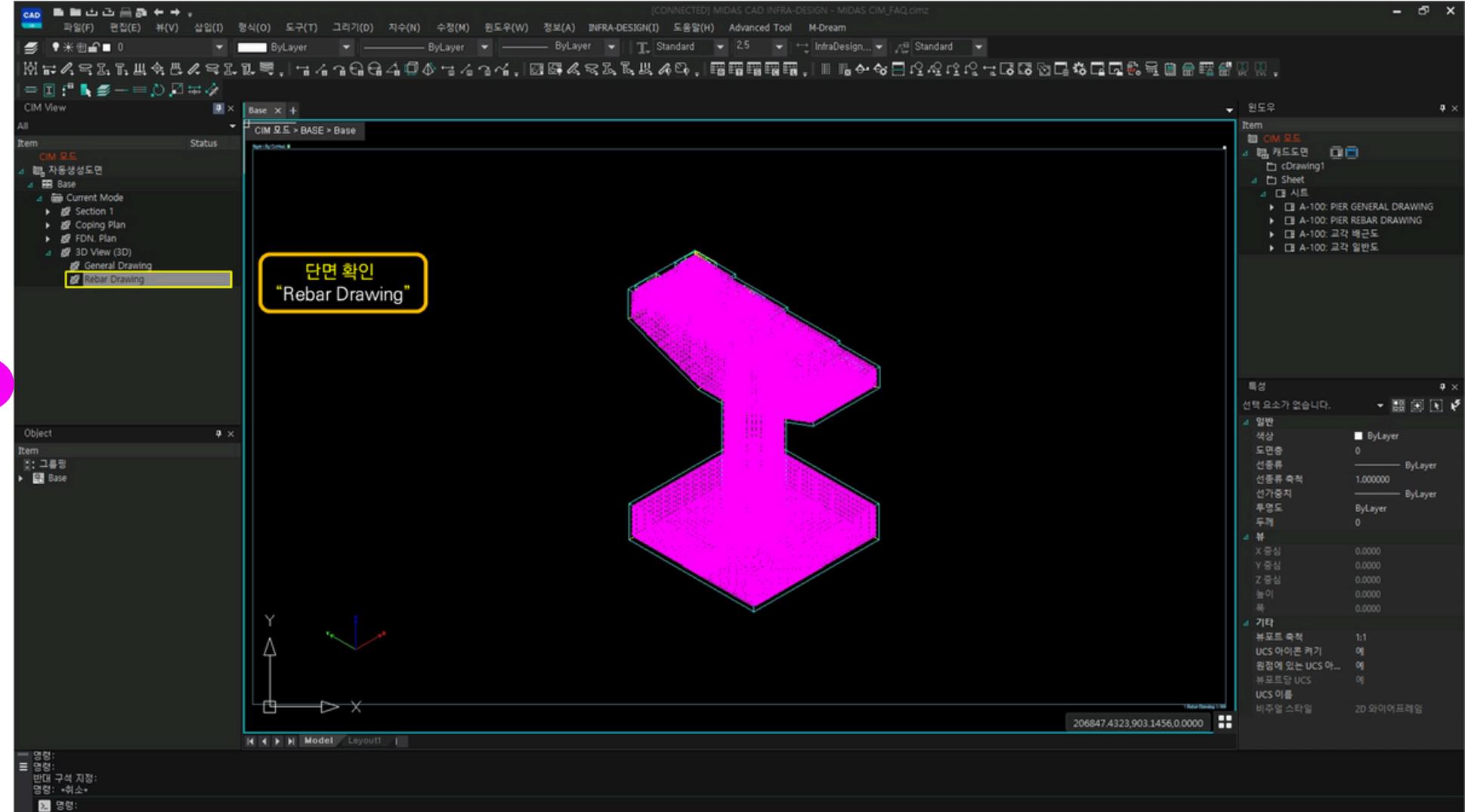
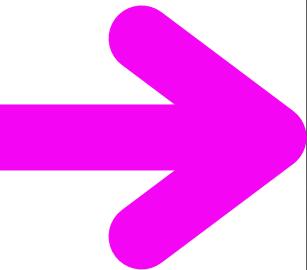
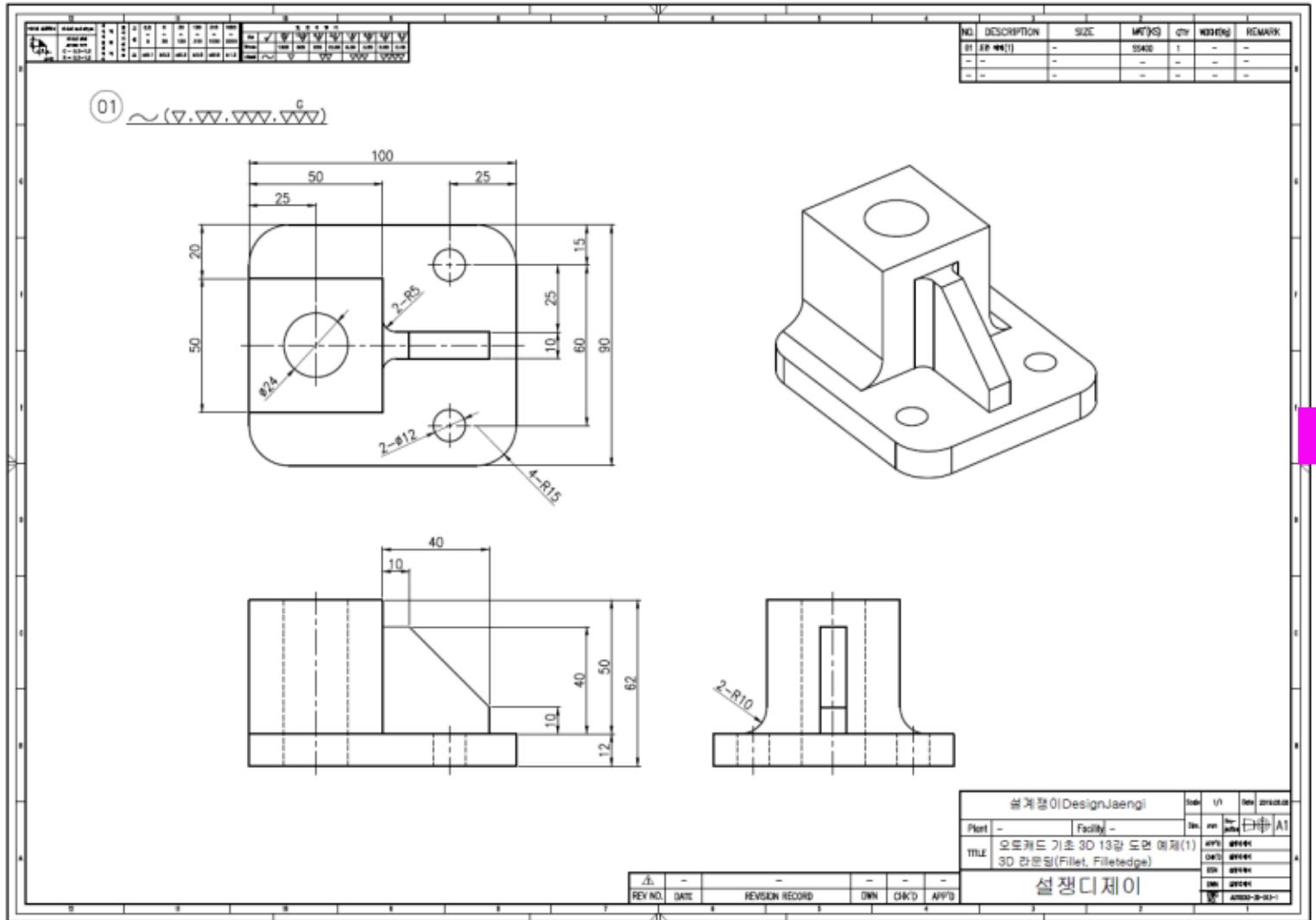
- [17] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. pages 216–224, 2018. [1](#), [2](#), [6](#), [7](#)
- [42] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018. [1](#), [2](#)
- [31] Charlie Nash, Yaroslav Ganin, S. M. Ali Eslami, and Peter Battaglia. PolyGen: An autoregressive generative model of 3D meshes. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7220–7229. PMLR, 13–18 Jul 2020. [1](#), [2](#), [4](#)

### Levelset files

- [12] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. [1](#), [2](#), [6](#), [7](#)
- [33] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. [1](#), [2](#)
- Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. [1](#), [2](#)
- [50] Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. Pq-net: A generative part seq2seq network for 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 829–838, 2020. [1](#), [2](#), [6](#)
- Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 45–54, 2020. [1](#), [2](#)

# 서론

## 문제 제기



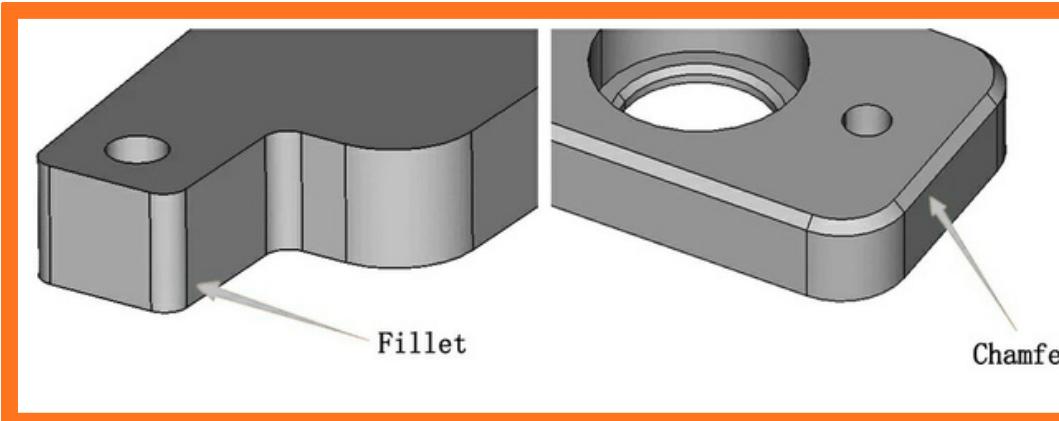
CAD를 통한 생성 모델은 매개변수적인 특성으로 인한 어려움이 있습니다.

CAD를 통한 생성 모델은 매개변수적인 특성으로 인한 어려움이 있습니다.

CAD 모델의 기하 연산(Curve sketch, extrusion, fillet, boolean, chamfer)으로 구성됩니다. 각 연산은 특정 매개변수에 의해 제어됩니다. 일부 매개변수는 불연속적인 옵션이고, 다른 매개변수는 연속적인 값을 갖습니다. 이런 불연속적인 점이 3D 생성에 있어 문제가 됩니다.

## 연구를 통한 해결 방법 제시

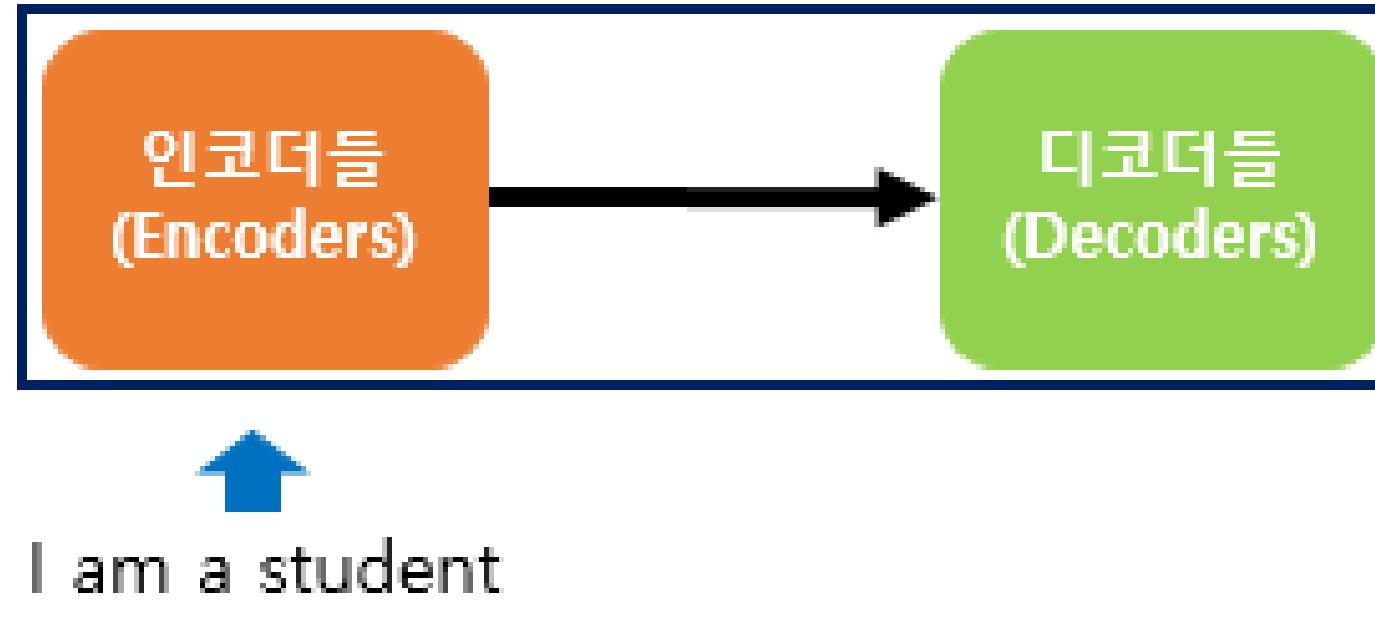
AutoCAD 명령어 [ Ver. 0729 2017 ]			
명령어[한글명]	명령어[단축키]	명령어[한글명]	명령어[단축키]
선	LINE-[L]	선 잇기	JOIN-[J]
지우기	ERASE-[E]	치수편집	DED
원	CIRCLE-[C]	길이조정	LENGTHEN-[LEN]
직사각형	RECTANG-[REC]	특성	PROPERTIES-[CH]
다각형	PLOYGON-[POL]	거리	DIST-[DI]
호	ARC-[A]	중첩 객체 삭제	OVERKILL
간격띄우기	OFFSET-[O]	무한대 선	XLINE-[XL]
모각기	FILLET-[F]	삽입>레스터 이미지	IMAGEATTACH-[IAT]
모따기	CHAMFER-[CHA]	레이어	LAYER-[LA]
자르기	TRIM-[TR]	신속지시선	QLEADER-[LE]
복사	COPY-[CO]	리스트	LIST-[LI]
이동	MOVE-[M]	선 종류축척	LSCALE-[LTS]
타원	ELLIPSE-[EL]	특성일치	MATCHPROP-[MA]
대칭	MIRROR-[MI]	도구>옵션	OPTION-[OP]



# 서론

je suis étudiant

## 트랜스포머 모델



코드 ^

**CYLINDER**

**Diameter** (옵션)

중심점 지정

반지름 또는 지름 입력

높이 입력

**Enter**

## CAD 명령어 시퀀스 데이터와 만드는 방법을 학습시킨 모델 제안

가장 자주 사용되는 CAD 작업을 고려한 명령 유형, 매개변수 및 순차적 순서를 인코딩 하기위해 공통 구조를 만듭니다. CAD 명령어 시퀀스를 자연어 처리를 위해, Transformer 네트워크에 자동 인코더를 제안. 이 인코더는 CAD 모델을 잠재 공간에 임베드하고, 나중에 잠재 벡터를 CAD 명령 시퀀스로 디코딩합니다.

# 기존 연구와 다른 점

# 서론

## 기존 연구에서 매개변수 형태 추론 방법 분류

### ParSeNet

- [38] Gopal Sharma, Difan Liu, Subhransu Maji, Evangelos Kalogerakis, Siddhartha Chaudhuri, and Radomír Měch. Parsenet: A parametric surface fitting network for 3d point clouds. In *European Conference on Computer Vision*, pages 261–276. Springer, 2020. [2](#)

3d Point Cloud를 매개변수를 표면 패치 세트를 받음.

### Pie-Net

- [43] Xiaogang Wang, Yuelang Xu, Kai Xu, Andrea Tagliasacchi, Bin Zhou, Ali Mahdavi-Amiri, and Hao Zhang. Pie-net: Parametric inference of point cloud edges. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20167–20178. Curran Associates, Inc., 2020. [2](#)

3d Point Cloud에서 매개변수를 경계 곡선을 추출해서 받습니다.

### Li et al

- [25] Changjian Li, Hao Pan, Adrien Bousseau, and Niloy J. Mitra. Sketch2cad: Sequential cad modeling by sketching in context. *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia 2020)*, 39(6):164:1–164:14, 2020. [2](#)

신경망에 합성 데이터를 학습시켜 결과물을 출력합니다.

### UV-Net, BrepNet

- [19] Pradeep Kumar Jayaraman, Aditya Sanghi, Joseph Lambourne, Thomas Davies, Hooman Shayani, and Nigel Morris. Uv-net: Learning from curve-networks and solids. *arXiv preprint arXiv:2006.10211*, 2020. [2](#)

- [24] Joseph G Lambourne, Karl DD Willis, Pradeep Kumar Jayaraman, Aditya Sanghi, Peter Meltzer, and Hooman Shayani. Brepnet: A topological message passing system for solid models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12773–12782, 2021. [2](#)

매개 변수로 경계 곡선과 표면 인코딩에 중점을 둡니다.

### Xu et al

- [51] Xianghao Xu, Wenzhe Peng, Chin-Yi Cheng, Karl DD Willis, and Daniel Ritchie. Inferring cad modeling sequences using zone graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6062–6070, 2021. [2](#)

neural-guided 를 적용하여 solid 모양에서 CAD 모델링 시퀀스를 추론

## 3D 생성 모델 분류

## 3D Point Cloud

- [6] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 40–49, Stockholm, Sweden, 10–15 Jul 2018. PMLR. 1, 2, 6, 7, 8, 4
- [52] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4541–4550, 2019. 1, 2
- [53] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2018. 1, 2
- [8] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 2
- [30] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. 2019. 1, 2

## Polygon Mesh

- [6] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 40–49, Stockholm, Sweden, 10–15 Jul 2018. PMLR. 1, 2, 6, 7, 8, 4
- [52] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4541–4550, 2019. 1, 2
- [53] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2018. 1, 2
- [8] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 2
- [30] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. 2019. 1, 2

## Voxel

- [49] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016. 2
- [16] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*, pages 484–499. Springer, 2016. 2
- [26] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)*, 36(4):to appear, 2017. 2
- [27] Yiyi Liao, Simon Donne, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2916–2925, 2018. 2

## Occupancy

- [12] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 1, 2, 6, 7
- [33] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 1, 2
- [29] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 1, 2
- [50] Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. Pq-net: A generative part seq2seq network for 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 829–838, 2020. 1, 2, 6
- [11] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 45–54, 2020. 1, 2

### 최근 향상된 3D 생성 모델 분류

# 서론

#### CSGNet

- [37] Gopal Sharma, Rishabh Goyal, Difan Liu, Evangelos Kalogerakis, and Subhransu Maji. Csgnet: Neural shape parser for constructive solid geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5515–5523, 2018. 2

복셀화된 형상 입력을 기반으로 Constructive Solid Geometry(CSG) 연산을 추론합니다.

#### UCSG-Net

- [21] Kacper Kania, Maciej Zięba, and Tomasz Kajdanowicz. Ucsg-net—unsupervised discovering of constructive solid geometry tree. *arXiv preprint arXiv:2006.09102*, 2020. 2

Constructive Solid Geometry(CSG) 트리에서 Supervision 없이 추론을 더욱 향상시켰습니다.

#### DSL

- [39] Yonglong Tian, Andrew Luo, Xingyuan Sun, Kevin Ellis, William T Freeman, Joshua B Tenenbaum, and Jiajun Wu. Learning to infer and execute 3d shape programs. *arXiv preprint arXiv:1901.02875*, 2019. 2

Homer Walke, R Kenny Jones, and Daniel Ritchie. Learning to infer shape programs using latent execution self training. *arXiv preprint arXiv:2011.13045*, 2020. 2

- [30] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. 2019. 1, 2

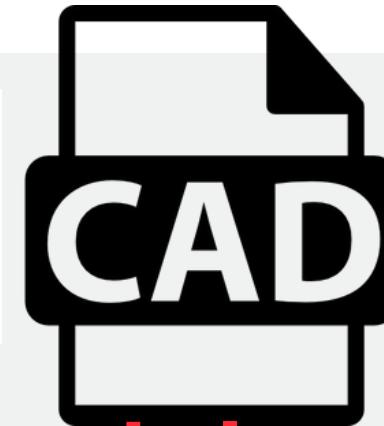
- [20] R. Kenny Jones, Theresa Barton, Xianghao Xu, Kai Wang, Ellen Jiang, Paul Guerrero, Niloy J. Mitra, and Daniel Ritchie. Shapcassembly: Learning to generate programs for 3d shape structure synthesis. *ACM Transactions on Graphics (TOG), Siggraph Asia 2020*, 39(6):Article 234, 2020. 2

DSL(Domain-Specific Language){특정 도메인의 문제를 해결하기 위해 특별히 설계된 프로그래밍 언어}

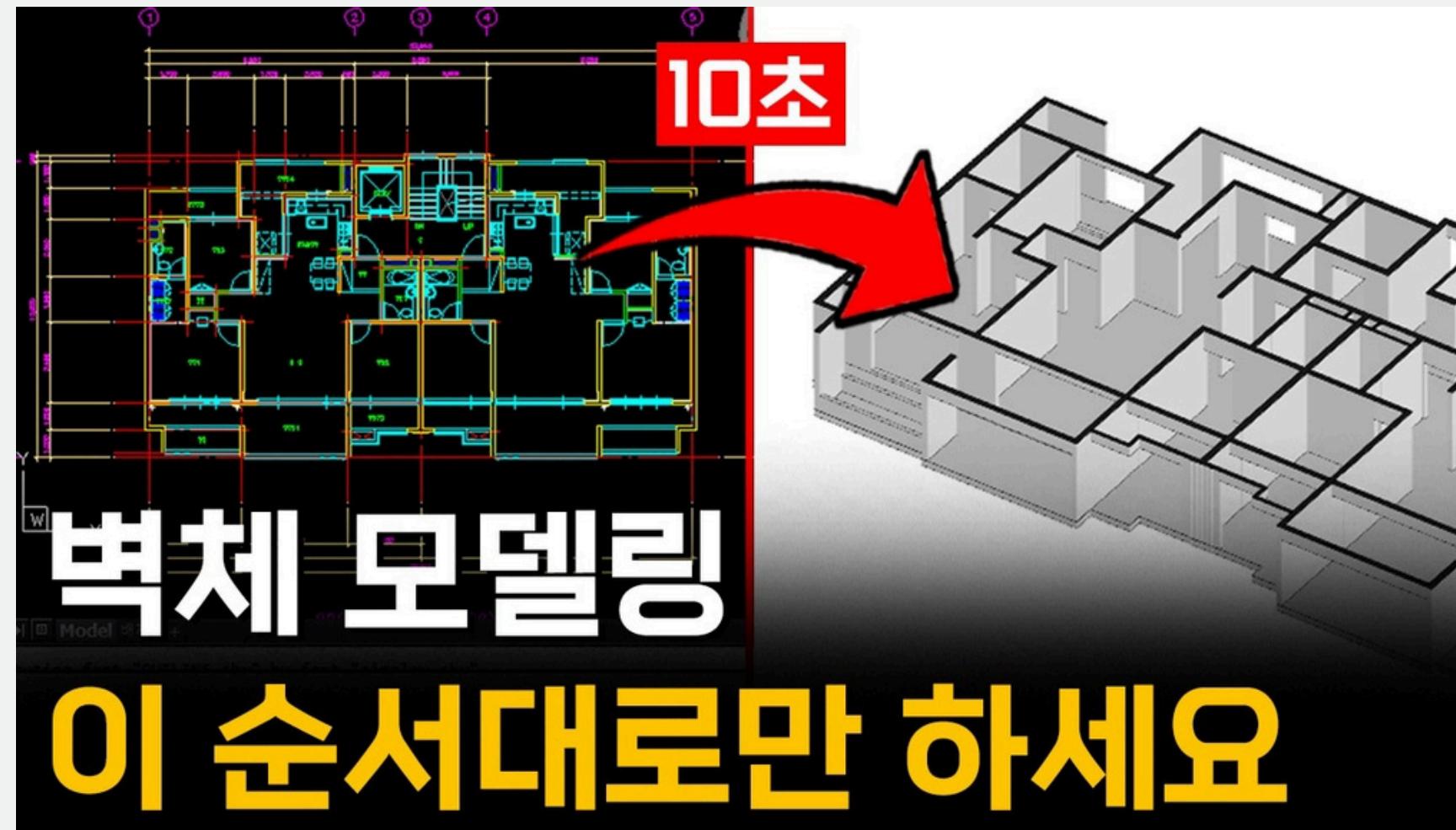
기존 연구와 다른 점  
본 연구

# 서론

- [1] Autocad. <https://www.autodesk.com/products/autocad>. 2
- [2] Fusion 360. <https://www.autodesk.com/products/fusion-360>. 2
- [3] Onshape. <http://onshape.com>. 2, 5



## TransFormer Model



프로그램을 통해 모델을 생성하고, 그것을 CAD 프로그램에 Import 해야 했던 과정없이,  
**CAD 프로그램 안에서 3D 모델 생성이 가능하게 만듭니다.**

# 서론

## Deep SVG

[10] Alexandre Carlier, Martin Danciljan, Alexandre Alahi, and Radu Timofte. Deepsvg: A hierarchical generative network for vector graphics animation. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16351–16361. Curran Associates, Inc., 2020. 2, 5

SVG(Scalable Vector Graphic) 이미지 생성을 위한 Transformer 기반 네트워크로 SVG 이미지는 선과 곡선과 같은 매개변수적 기본 요소의 집합으로 표현됩니다. 2차원이라는 것을 제외하고, 특정 순서나 종속성 없이 그룹화됩니다.

## 본 연구 Deep CAD

CAD 명령은 3차원으로 표현됩니다. 이러한 명령은 종속적일 수도 있으면 특정 순서를 따라야 합니다.

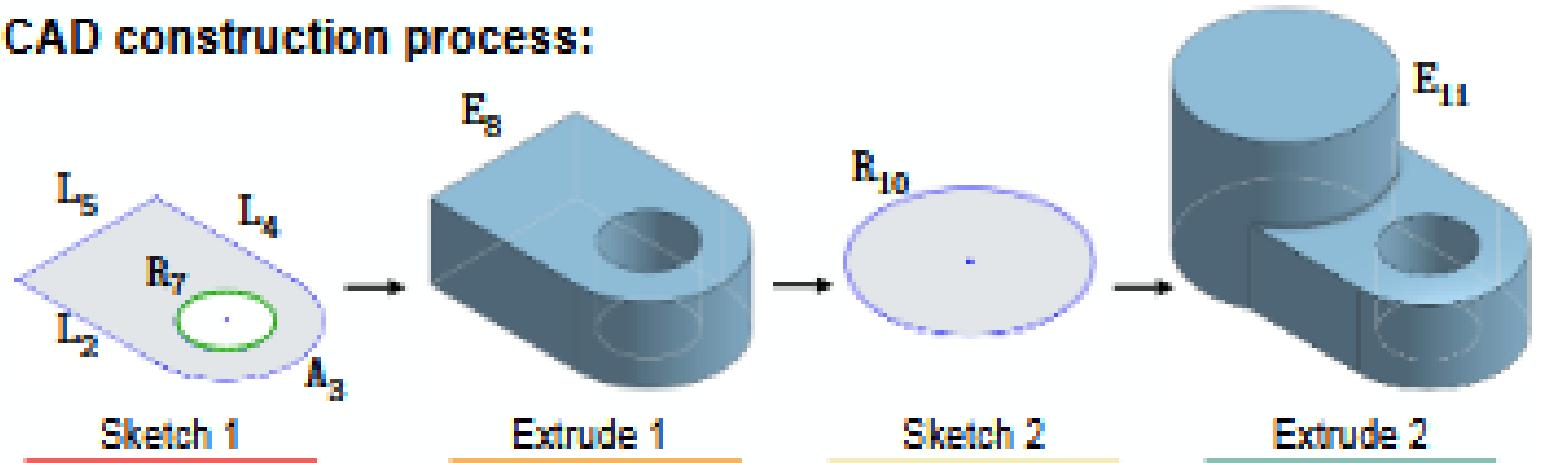
따라서 본 연구에서는 Transformer 기반 자동 인코더에서 CAD 명령과 그 순차적 순서를 인코딩 하는 새로운 방법을 알아봅니다.

# 본론

## 연구 방법

### CAD Representation for Neural Networks

#### CAD construction process:



#### Parametrized command sequence:

$\langle \text{SOL} \rangle_1 : \emptyset$	$E_8 : (0, 0, 0, -2, -1, 0, 3,$
$L_2 : (2, 0)$	$1, 0, \text{New body}, \text{One-sided})$
$A_3 : (2, 2, \pi, 1)$	$\langle \text{SOL} \rangle_9 : \emptyset$
$L_4 : (0, 2)$	$R_{10} : (0, 0, 1.125)$
$L_5 : (0, 0)$	$E_{11} : (0, 0, 0, -2, 0, 0, 2.25,$
$\langle \text{SOL} \rangle_6 : \emptyset$	$2, 0, \text{Join}, \text{One-sided})$
$R_7 : (2, 1, 0.5)$	$\langle \text{EOS} \rangle_{12} : \emptyset$

1. Sketch 1에서 회색으로 표시된 2D 면에서 L2-A3-L4-L5(파란색)는 면을 생성합니다.
2. R7(초록색)는 또 다른 면을 생성합니다.

1. 2D 평면에 곡선의 면을 만든 다음, 이를 3D로 만듭니다.
2. 만들어진 도형과 다른 3D 모양과 합집합 같은 연산을 통해 처리됩니다. 이러한 명령을 CAD 명령 시퀀스라고 합니다.
3. CAD 명령 시퀀스 뒤에는 모델의 경계 표현(B-rep)으로 알려진 표현이 있습니다.
4. 명령 시퀀스가 제공되면 B-rep은 자동 계산됩니다. B-rep은 위산 구성 요소(정점, 매개변수 모서리 및 면)와 이들 간의 연결로 구성됩니다.
5. 본 연구에서는 B-rep이 아닌 CAD 명령 시퀀스의 생성 모델 목표로 합니다. 명령 시퀀스는 B-rep으로 쉽게 변환할 수 있지만, 그 반대는 어렵기 때문입니다.

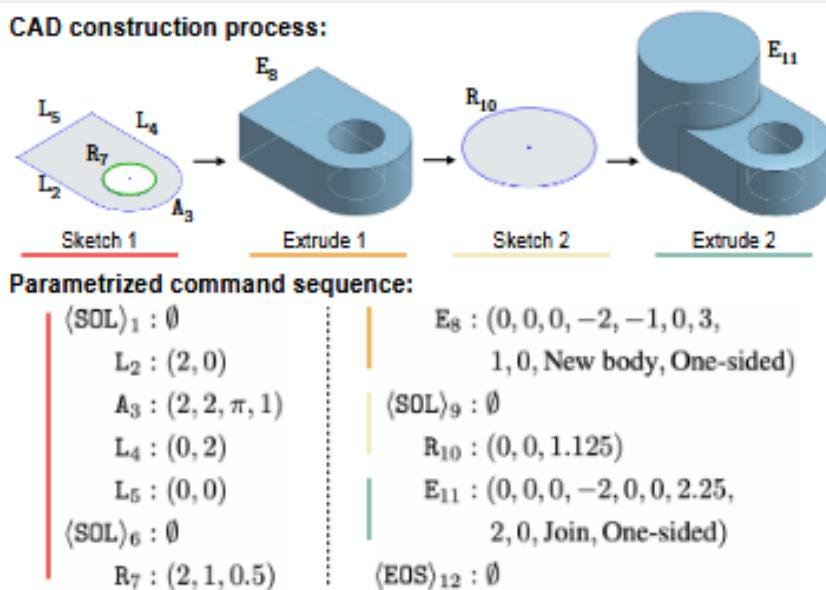
# 본론

## 연구 방법

### Specification of CAD Commands

Commands	Parameters
<code>(SOL)</code>	$\emptyset$
<code>L</code> (Line)	$x, y$ : line end-point
<code>A</code> (Arc)	$x, y$ : arc end-point $\alpha$ : sweep angle $f$ : counter-clockwise flag
<code>R</code>	$x, y$ : center
<code>(Circle)</code>	$r$ : radius $\theta, \phi, \gamma$ : sketch plane orientation
<code>E</code> (Extrude)	$p_x, p_y, p_z$ : sketch plane origin $s$ : scale of associated sketch profile $e_1, e_2$ : extrude distances toward both sides $b$ : boolean type, $u$ : extrude type
<code>(EOS)</code>	$\emptyset$

Table 1. CAD commands and their parameters. `(SOL)` indicates the start of a loop; `(EOS)` indicates the end of the whole sequence.



In short, a sketch profile  $S$  is described by a list of loops  $S = [Q_1, \dots, Q_N]$ , where each loop  $Q_i$  consists of a series of curves starting from the indicator command `(SOL)` (i.e.,  $Q_i = [(\text{SOL}), C_1, \dots, C_{n_i}]$ ), and each curve command  $C_j = (t_j, p_j)$  specifies the curve type  $t_j$  and its shape parameters  $p_j$  (see Fig. 2).

CAD 명령어는 자주 쓰는 명령어가 정해져 있습니다. 왼쪽 그림과 같이 명령어를 정리합니다.

이 논문에서 용어 정리로, 닫힌 곡선은 loop(루프)라고 합니다. 하나 이상의 loop(루프)는 profile(프로파일)이라는 닫힌 영역을 만듭니다.

본 연구에서는 Profile(프로파일)을 경계에 있는 loop(루프) 목록으로 표현합니다.

루프는 항상 지시자 명령`<SOL>`으로 시작합니다. 그 뒤에 곡선 명령  $C_i$ 가 이어집니다.

루프의 모든 곡선은 시작점이 가장 왼쪽 아래에 있는 곡선부터 시작하여, 시계 반대 방향으로 나열됩니다.

Profile(프로파일)의 loop(루프)는 경계 상자의 왼쪽 아래 모서리를 기준으로 정렬됩니다.

# 본론

## 연구 방법

### Specification of CAD Commands

In short, a sketch profile  $S$  is described by a list of loops  $S = [Q_1, \dots, Q_N]$ , where each loop  $Q_i$  consists of a series of curves starting from the indicator command  $\langle \text{SOL} \rangle$  (i.e.,  $Q_i = [\langle \text{SOL} \rangle, C_1, \dots, C_{n_i}]$ ), and each curve command  $C_j = (t_j, p_j)$  specifies the curve type  $t_i$  and its shape parameters  $p_j$  (see Fig. 2).

각 곡선 명령  $C_i$ 는 곡선 유형  $i \in \{\langle \text{SOL} \rangle, L, A, R\}$ 과 표 1에 나열된 매개변수로 설명됩니다.  
곡선 매개변수는 스케치 평면의 로컬 기준 프레임에서 곡선의 2D 위치를 지정합니다.

Extrusion 명령어

Extrusion 명령어 사용방법

2D 평면에서 3D를 만들어냅니다. 단면, 양면 일 수 있습니다. 3D 본체를 이전에 생성된 3D 모델과 병합하는 방법을 bool 연산 중 하나를 통해 지정합니다. 본체와 결합 혹은 절단하는 것입니다.

2D 평면을 3D 방향으로 정의해야 합니다. 이는 Table1의  $(\theta, \gamma, \varphi)$  매개변수로 결정되는 회전 행렬에 의해 결정됩니다.  
이 행렬은 월드 기준 Z축을 평면의 법선 방향에 정렬합니다. 2D 평면에서 3D로 얼마나 둘출될지 결정하는 축척 계수  $s$ 가 포함됩니다.

# 본론

## 연구 방법

### Network-friendly Representation

CAD 명령어는 자연어와 다릅니다. 각 명령은 서로 다른 수의 매개변수를 갖습니다.  
일부 명령에서는 매개변수가 연속 값과 이산 값이 혼합되어 있습니다.

**명령 시퀀스는 신경망에서 직접 사용하기에 적합하지 않습니다.**

이것을 극복하기 위해 명령 시퀀스의 차원을 정규화합니다.  
먼저, 각 명령의 매개변수를  $16 \times 1$  벡터로 쌓아 올립니다.  
이 벡터의 요소는 표에 적힌 모든 명령의 집한 매개변수에 해당합니다.  
 $p_i = [x, y, \alpha, f, r, \theta, \varphi, y, px, py, pz, s, e1, e2, b, u]$ . 각 명령에 대해 사용되지 않는 매개변수는 간단히 -1로 설정합니다.

모든 CAD 모델 M의 총 명령 수  $N_c$ 를 고정합니다. 이는 시퀀스 길이가  $N_c$ 에 도달할 때까지 CAD 모델의 명령 시퀀스에 빈 명령  $\langle EOS \rangle$  를 채워 넣는 방식으로 수행됩니다. 실제로는 훈련 데이터 세트에 나타난 최대 명령 시퀀스 길이인  $N_c = 60$ 을 선택합니다.

매개변수를 양자화하여 연속 및 불연속 매개변수를 통합합니다. 이를 위해  $2 \times 2 \times 2$  큐브 내의 모든 CAD 모델을 정규화합니다.  
축척 계수  $s$ 를 포함하여 정규화된 프로파일을 원래 크기로 복원합니다.  
정규화는 연속 매개변수의 범위를 제한하여, 값을 256단계로 양자화하고 8비트 정수를 사용하여 표현할 수 있도록 합니다.

이 부분은 모델을 최적화를 위하여, 양자화를 한다는 글이라고 생각됩니다. 저도 정확한 의미를 해석하진 못했습니다.

# 본론

## 연구 방법

### Network-friendly Representation

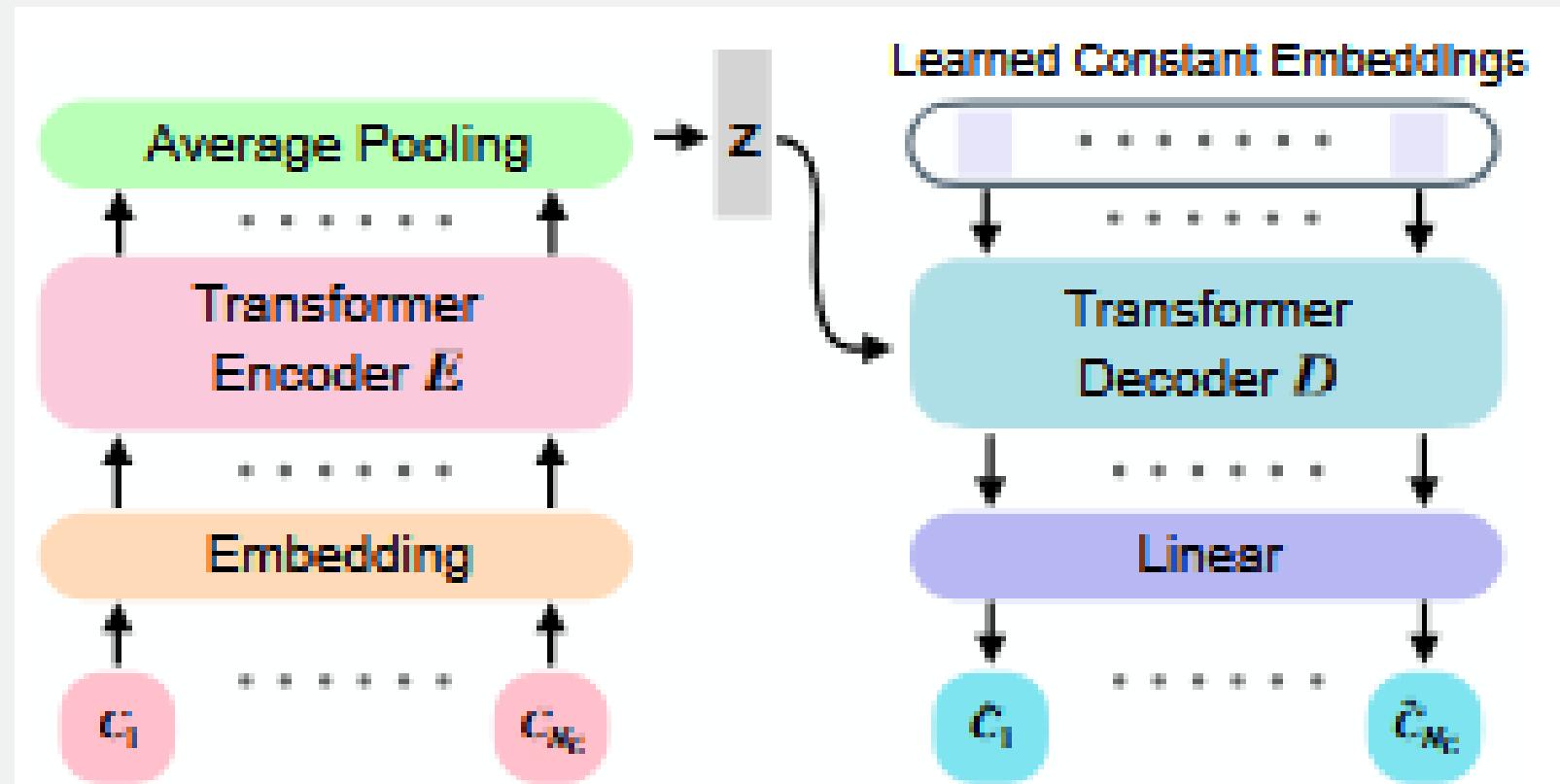
CAD 설계에서는 평행 및 수직 스케치 선과 같은 특정 기하학적 관계를 준수해야 합니다. 그러나 생성 모델이 연속 매개변수를 직접 생성하는 경우, 매개변수 회귀를 통해 얻은 값은 이러한 엄격한 관계를 깨는 오류가 발생하기 쉽습니다.

대신 매개변수 양자화를 통해 네트워크는 매개변수를 특정 수준으로 "분류"하여 학습된 기하학적 관계를 더 잘 준수할 수 있습니다.

# 본론

## 연구 방법

### Autoencoder for CAD Models



CAD 명령에 표현을 활용하는 자동 인코더 네트워크 모델 구조입니다.

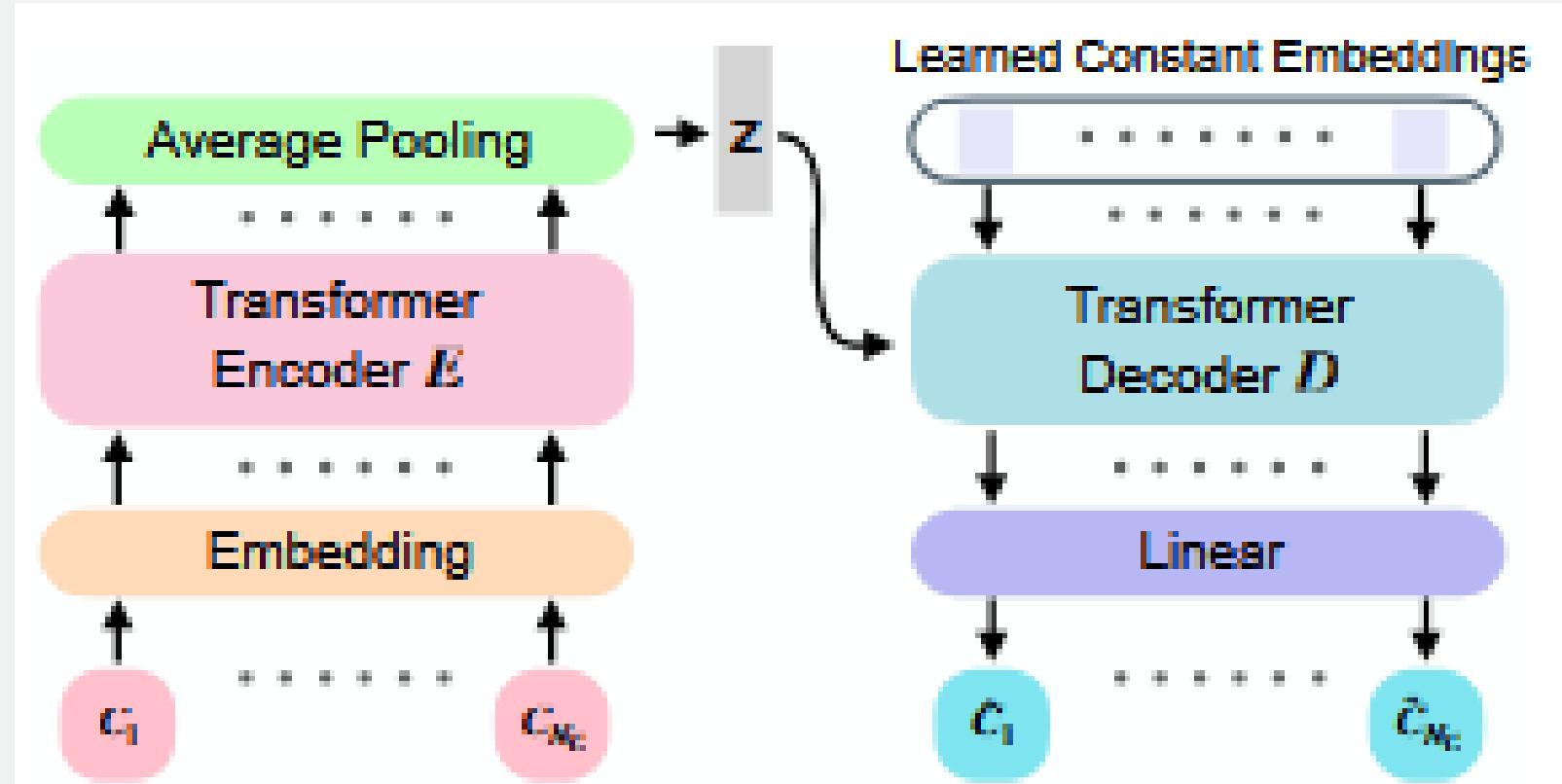
자동 인코더는 CAD 명령 시퀀스  $M = [C_1, \dots, C_{N_c}]$ 를 입력으로 받습니다. 여기서  $N_c$ 는 고정된 수입니다.

각 명령  $C_i$ 는 차원  $d_E = 256$ 의 연속 임베딩 공간에 개별적으로 투영됩니다. 그런 다음 모든 임베딩을 합하여 인코더  $E$ 에 입력하면, 인코더  $E$ 는 잠재 벡터  $z \in \mathbb{R}^{256}$ 을 출력합니다. 디코더는 잠재 벡터  $z$ 를 입력으로 받고 생성된 CAD 명령 시퀀스  $\hat{M}$ 을 출력합니다.

# 본론

## 연구 방법

### Embedding



CAD 명령어  $C_i = (t_i, p_i)$ 는 명령어 유형  $t_i$ 와 매개변수  $p_i$ 라는 두 가지 뚜렷한 부분을 갖습니다.  
 $C_i$ 의 임베딩을 계산하는 다른 방법을 공식화합니다.

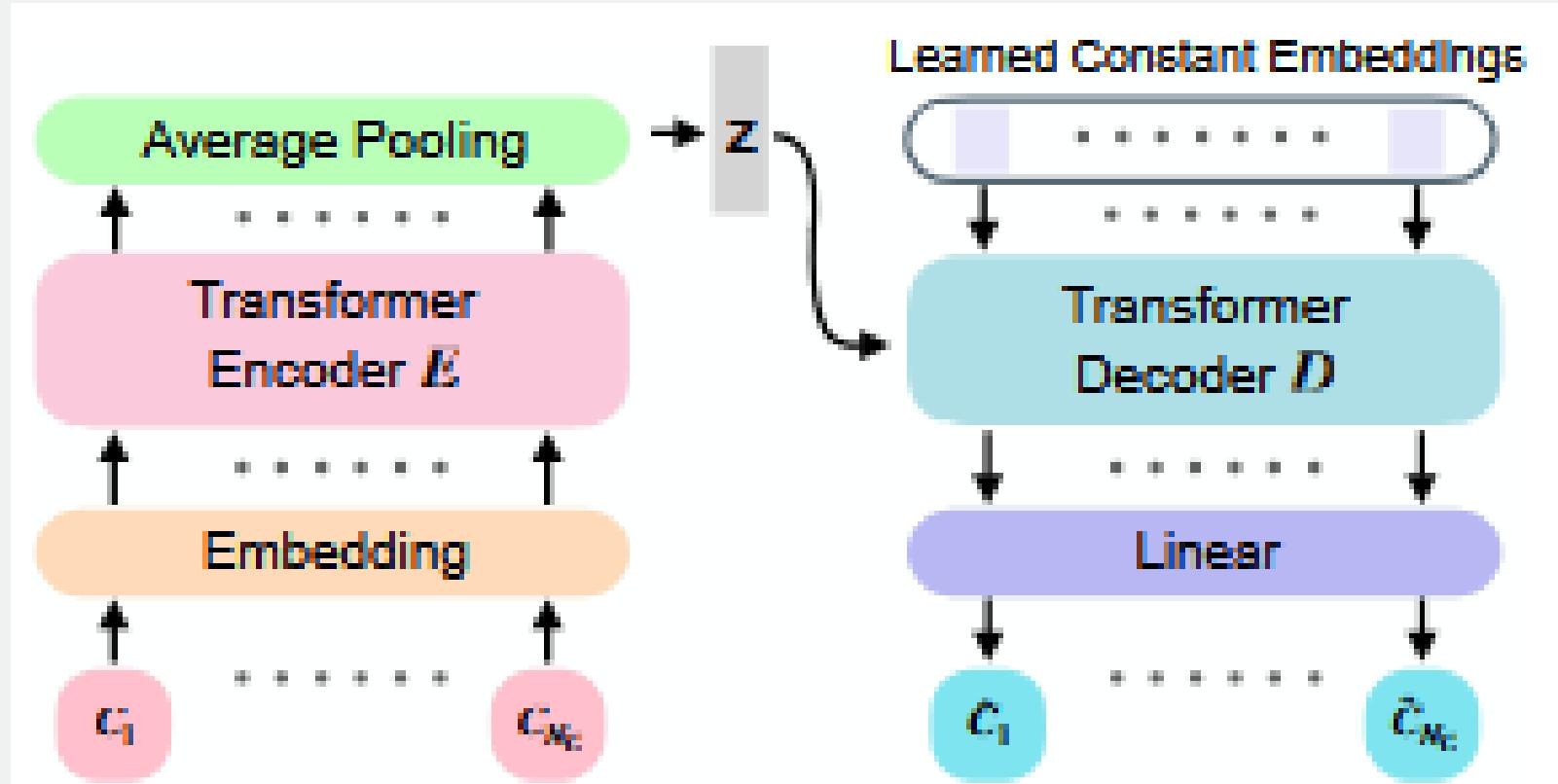
$$e(C_i) = \text{ecmd } i + \text{eparam } i + \text{epos } i \in \mathbb{R}^{d_E}$$
 입니다.

첫 번째 임베딩  $\text{ecmd } i$ 는  $\text{ecmd } i = W_{\text{cmd}} \delta c_i$ 로 주어진 명령어 유형  $t_i$ 를 고려합니다. 여기서  $W_{\text{cmd}} \in \mathbb{R}^{d_E \times 6}$ 은 학습 가능한 행렬이고  $\delta c_i \in \mathbb{R}^6$ 은 6가지 명령어 유형 중 명령어 유형  $t_i$ 를 나타내는 원핫 벡터입니다.

# 본론

## 연구 방법

### Embedding



$$e_i^{\text{param}} = W_{\text{param}}^a \text{flat}(W_{\text{param}}^b \delta_i^p),$$

두 번째 임베딩  $e_{\text{param}}^i$ 는 명령 매개변수를 고려합니다.

모든 명령은 16개의 매개변수를 가지며, 각 매개변수는 8비트 정수로 양자화됩니다. 이 정수 각각을 차원  $28 + 1 = 257$ 의 원-핫 벡터  $\delta_{i,j}$  ( $j = 1..16$ )로 변환합니다.

추가 차원은 해당 매개변수가 해당 명령에서 사용되지 않음을 나타냅니다. 모든 원-핫 벡터를 행렬  $\delta_p$

$i \in \mathbb{R}^{257 \times 16}$ 에 쌓고, 또 다른 학습 가능 행렬  $W_b$

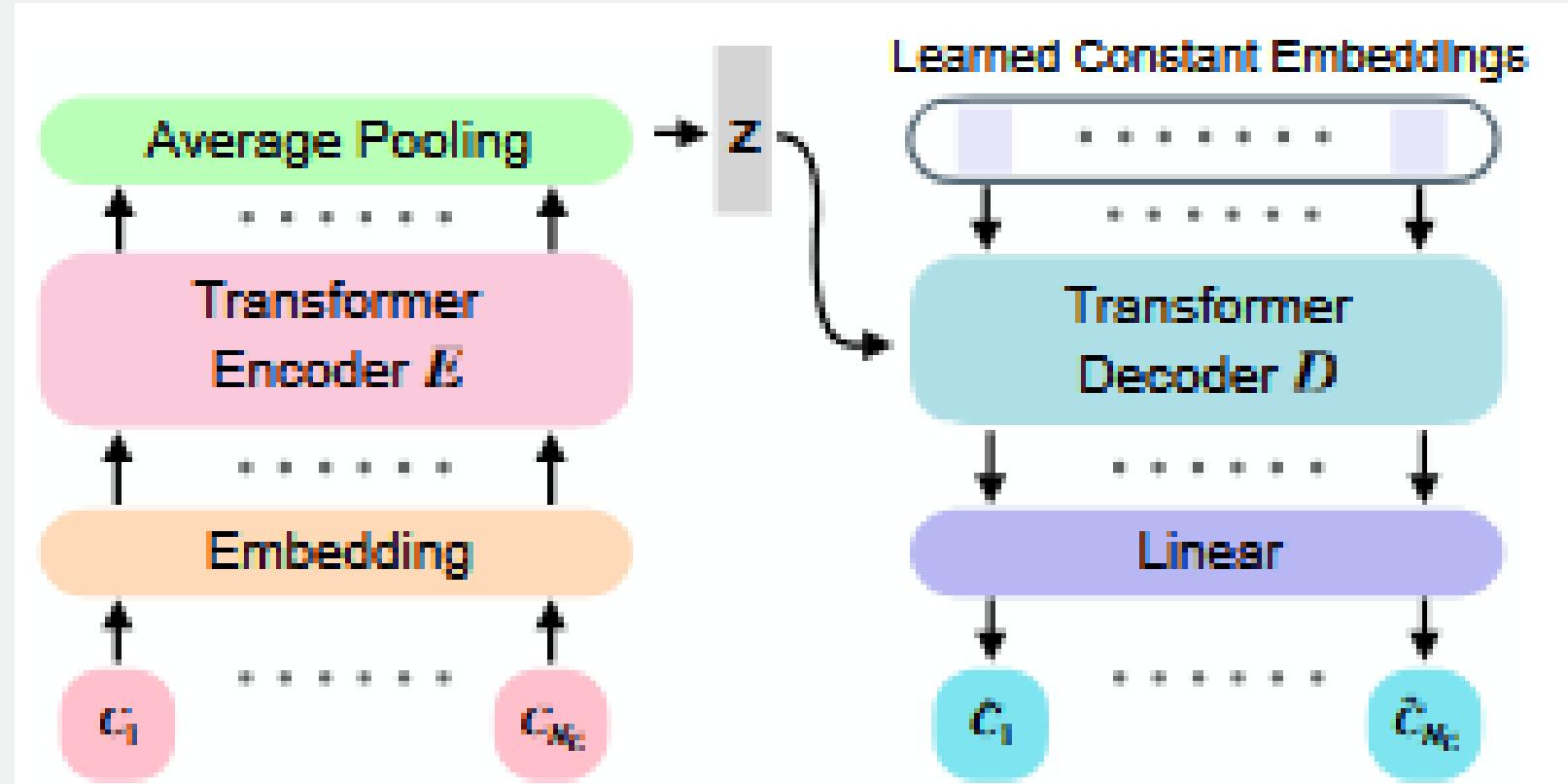
$\text{param} \in \mathbb{R}^{d_E \times 257}$ 을 사용하여 각 매개변수를 개별적으로 임베딩합니다. 그런 다음 선형 계층  $W_a$

$\text{param} \in \mathbb{R}^{d_E \times 16d_E}$ 를 통해 개별 임베딩을 결합합니다.

# 본론

## 연구 방법

### Embedding



$$e_i^{\text{param}} = W_{\text{param}}^a \text{flat}(W_{\text{param}}^b \delta_i^p),$$

flat( $\cdot$ )은 입력 행렬을 벡터로 평탄화합니다.

마지막으로, [40]과 유사하게 위치 임베딩  $e_{\text{pos}}$

$i$ 는

전체 명령 시퀀스에서 명령  $C_i$ 의 인덱스를 나타냅니다.

$e_{\text{pos}}$

$i = W_{\text{pos}} \delta_i$ 로 정의되며, 여기서  $W_{\text{pos}} \in R^{d_E \times N_c}$ 는 학습 가능한 행렬이고  $\delta_i \in R^{N_c}$ 는 인덱스  $i$ 에서 1로 채워지고 그렇지 않으면 0으로 채워진 원핫 벡터입니다.

[여기 부분은 이해하지 못했습니다.]

# 본론

## 연구 방법

### Creation of CAD Dataset

[23] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 5, 1

[3] Onshape. <http://onshape.com>. 2, 5

Karl D. D. Willis, Yewen Pu, Jieliang Luo, Hang Chu, Tao Du, Joseph G. Lambourne, Armando Solar-Lezama, and Wojciech Matusik. Fusion 360 gallery: A dataset and environment for programmatic cad reconstruction. *arXiv preprint arXiv:2010.02392*, 2020. 3, 5, 7, 1

데이터 셋을 구하는게 쉽지 않습니다.

그래서 CAD 명령 시퀀스를 제공하는 새로운 데이터 세트를 만들었습니다. 이 데이터 셋은 향후 연구에도 활용할 수 있습니다.

해당 데이터 셋은 링크를 첨부합니다.

각 CAD 모델에 대한 데이터 셋트는 Onshape의 도메인별 언어를 사용하여 해당 설계에 사용된 CAD 작업과 매개변수를 구문 분석합니다.

스케치 및 생성 이외의 작업을 사용하는 CAD 모델은 삭제합니다. 나머지 모델의 경우 FeatureScript 프로그램을 사용하여 명령어에 적인 표현을 사용하여 만듭니다.

# 본론

## 연구 방법

### Creation of CAD Dataset

[23] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 5, 1

[3] Onshape. <http://onshape.com>. 2, 5

Karl D. D. Willis, Yewen Pu, Jieliang Luo, Hang Chu, Tao Du, Joseph G. Lambourne, Armando Solar-Lezama, and Wojciech Matusik. Fusion 360 gallery: A dataset and environment for programmatic cad reconstruction. *arXiv preprint arXiv:2010.02392*, 2020. 3, 5, 7, 1

데이터 셋을 구하는게 쉽지 않습니다.

그래서 CAD 명령 시퀀스를 제공하는 새로운 데이터 세트를 만들었습니다. 이 데이터 셋은 향후 연구에도 활용할 수 있습니다.

해당 데이터 셋은 링크를 첨부합니다.

각 CAD 모델에 대한 데이터 셋트는 Onshape의 도메인별 언어를 사용하여 해당 설계에 사용된 CAD 작업과 매개변수를 구문 분석합니다.

스케치 및 생성 이외의 작업을 사용하는 CAD 모델은 삭제합니다. 나머지 모델의 경우 FeatureScript 프로그램을 사용하여 명령어에 적인 표현을 사용하여 만듭니다.

178,238개의 CAD 설계를 포함하는 데이터 세트를 만들었습니다

# 본론

## 연구 방법

### Training and Runtime Generation

$$\mathcal{L} = \sum_{i=1}^{N_c} \ell(\hat{t}_i, t_i) + \beta \sum_{i=1}^{N_c} \sum_{j=1}^{N_P} \ell(\hat{p}_{i,j}, p_{i,j}),$$

예측된 CAD 모델  $\hat{M}$ 과  
실제 모델  $M$  간의 손실을 위와 같이 정의합니다.

여기서  $\ell(\cdot, \cdot)$ 는 표준 교차 엔트로피를 나타내고,  $N_p$ 는 매개변수 수(예시에서는  $N_p = 16$ )이며,  $\beta$ 는 두 항의 균형을 맞추는 가중치(예시에서는  $\beta = 2$ )입니다.  
실제 명령 시퀀스에서 일부 명령은 비어 있고(예시에서는 패딩 명령  $\langle \text{EOS} \rangle$ ) 일부 명령 매개변수는 사용되지 않습니다.

학습 과정은 Adam 최적화기[22]를 사용하며, 학습률은 0.001이고 선형 워밍업 기간은 초기 단계 2000개입니다. 모든 Transformer 블록에 대해 드롭아웃 비율을 0.1로 설정하고 역전파에 그래디언트 클리핑을 1.0으로 적용합니다. 배치 크기는 512로 1000 에포크 동안 네트워크를 학습합니다.

# 본론

## 연구 방법

### Training and Runtime Generation

$$\mathcal{L} = \sum_{i=1}^{N_c} \ell(\hat{t}_i, t_i) + \beta \sum_{i=1}^{N_c} \sum_{j=1}^{N_P} \ell(\hat{p}_{i,j}, p_{i,j}),$$

오토인코더가 잘 학습되면

256차원 잠재 벡터  $z$ 를 사용하여 CAD 모델을 표현할 수 있습니다. CAD 모델의 자동 생성을 위해 학습된 잠재 공간에 잠재 GAN 기법[6, 12, 50]을 적용합니다. 생성기와 판별기는 모두

4개의 은닉층을 가진 다층 퍼셉트론(MLP) 네트워크만큼 간단하며,

그래디언트 페널티를 적용한 Wasserstein-GAN 학습 전략[7, 18]을 사용하여 학습합니다. 마지막으로

CAD 모델을 생성하기 위해 다변량 가우시안 분포에서 랜덤 벡터를 샘플링하여

GAN 생성기에 입력합니다. GAN의 출력은 잠재 벡터  $z$ 이며, 이는 Transformer 기반 디코더에 입력됩니다.

여기는 잘 이해하지 못했습니다.

# 본론

## 연구 방법

### Metrics 지표

$$\text{ACC}_{\text{cmd}} = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbb{I}[t_i = \hat{t}_i].$$

$$\text{ACC}_{\text{param}} = \frac{1}{K} \sum_{i=1}^{|p_i|} \sum_{j=1}^{N_c} \mathbb{I}[|p_{i,j} - \hat{p}_{i,j}| < \eta] \mathbb{I}[t_i = \hat{t}_i], \quad (5)$$

성능을 측정하기 위해서 명령 정확도(ACCcmd)와 매개변수 정확도(ACCparam)이라는 두 가지 지표를 사용하여 명령 정확도를 평가합니다.

# 본론

## 연구 방법

### Metrics 지표

$$\text{ACC}_{\text{cmd}} = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbb{I}[t_i = \hat{t}_i].$$

$$\text{ACC}_{\text{param}} = \frac{1}{K} \sum_{i=1}^K \sum_{j=1}^{|p_i|} \mathbb{I}[|p_{i,j} - \hat{p}_{i,j}| < \eta] \mathbb{I}[t_i = \hat{t}_i], \quad (5)$$

성능을 측정하기 위해서 명령 정확도(ACCcmd)와 매개변수 정확도(ACCparam)이라는 두 가지 지표를 사용하여 명령 정확도를 평가합니다.

복구된 3D 지오메트리의 품질을 측정하기 위해

이전의 여러 이산화된 형상(예: 포인트 클라우드) 생성 모델에서 사용된 지표인 챔퍼 거리(CD)를 사용합니다. 여기서는 기준 형상과 복구된 형상의 표면에서 각각 2000개의 점을 균일하게 샘플링하여 CD를 평가하고, 두 점 집합 사이의 CD를 측정합니다.

출력 CAD 명령 시퀀스가 항상 유효한 3D 형상을 생성한다는 보장은 없습니다. 드물지만 출력 명령으로 인해 잘못된 토폴로지가 생성되어 해당 CAD 모델에서 포인트 클라우드를 추출할 수 없는 경우가 있습니다. 따라서 포인트 클라우드로 변환되지 못한 출력 CAD 모델의 비율인 무효 비율도 함께 평가합니다.

# 본론

## 연구 방법

### Comparison methods

Method	mean	trimmed mean	median
	CD	CD	CD
Ours+Aug	<b>6.14</b>	<b>0.974</b>	<b>0.752</b>
Ours	7.16	1.08	0.787
Alt-ArcMid	6.90	1.09	0.790
Alt-Trans	7.14	1.09	0.792
Alt-Rel	9.24	1.38	0.863
Alt-Regr	12.61	3.87	2.14

Alt-Rel은 루프에서 이전 곡선의 위치를 기준으로 곡선 위치를 나타냅니다. 이는 곡선 지정 시 절대 위치를 사용하는 본 모델과 대조됩니다.

Alt-Trans는 돌출 명령에 루프의 시작점 위치(스케치 평면의 원점 외에도)를 포함합니다. 여기서 시작점 위치와 평면의 원점은 CAD 모델의 월드 좌표계에 있습니다. 이와 대조적으로, 제안하는 방법은 스케치 평면의 원점만 포함하고 원점은 루프의 시작 위치로 변환됩니다. 따라서 더 간결합니다. Alt-ArcMid는 끝점과 중간점 위치를 사용하여 호를 지정하지만, 표 1에 사용된 스위핑 각도와 반시계 방향 플래그는 지정하지 않습니다.

Alt-Regr은 손실 함수의 표준 평균 제곱 오차를 사용하여 CAD 명령의 모든 매개변수를 회귀합니다.

# 본론

## 연구 방법

### Comparison methods

Method	mean	trimmed mean	median
	CD	CD	CD
Ours+Aug	<b>6.14</b>	<b>0.974</b>	<b>0.752</b>
Ours	7.16	1.08	0.787
Alt-ArcMid	6.90	1.09	0.790
Alt-Trans	7.14	1.09	0.792
Alt-Rel	9.24	1.38	0.863
Alt-Regr	12.61	3.87	2.14

Alt-Rel은 루프에서 이전 곡선의 위치를 기준으로 곡선 위치를 나타냅니다. 이는 곡선 지정 시 절대 위치를 사용하는 본 모델과 대조됩니다.

Alt-Trans는 돌출 명령에 루프의 시작점 위치(스케치 평면의 원점 외에도)를 포함합니다. 여기서 시작점 위치와 평면의 원점은 CAD 모델의 월드 좌표계에 있습니다. 이와 대조적으로, 제안하는 방법은 스케치 평면의 원점만 포함하고 원점은 루프의 시작 위치로 변환됩니다. 따라서 더 간결합니다. Alt-ArcMid는 끝점과 중간점 위치를 사용하여 호를 지정하지만, 표 1에 사용된 스위핑 각도와 반시계 방향 플래그는 지정하지 않습니다.

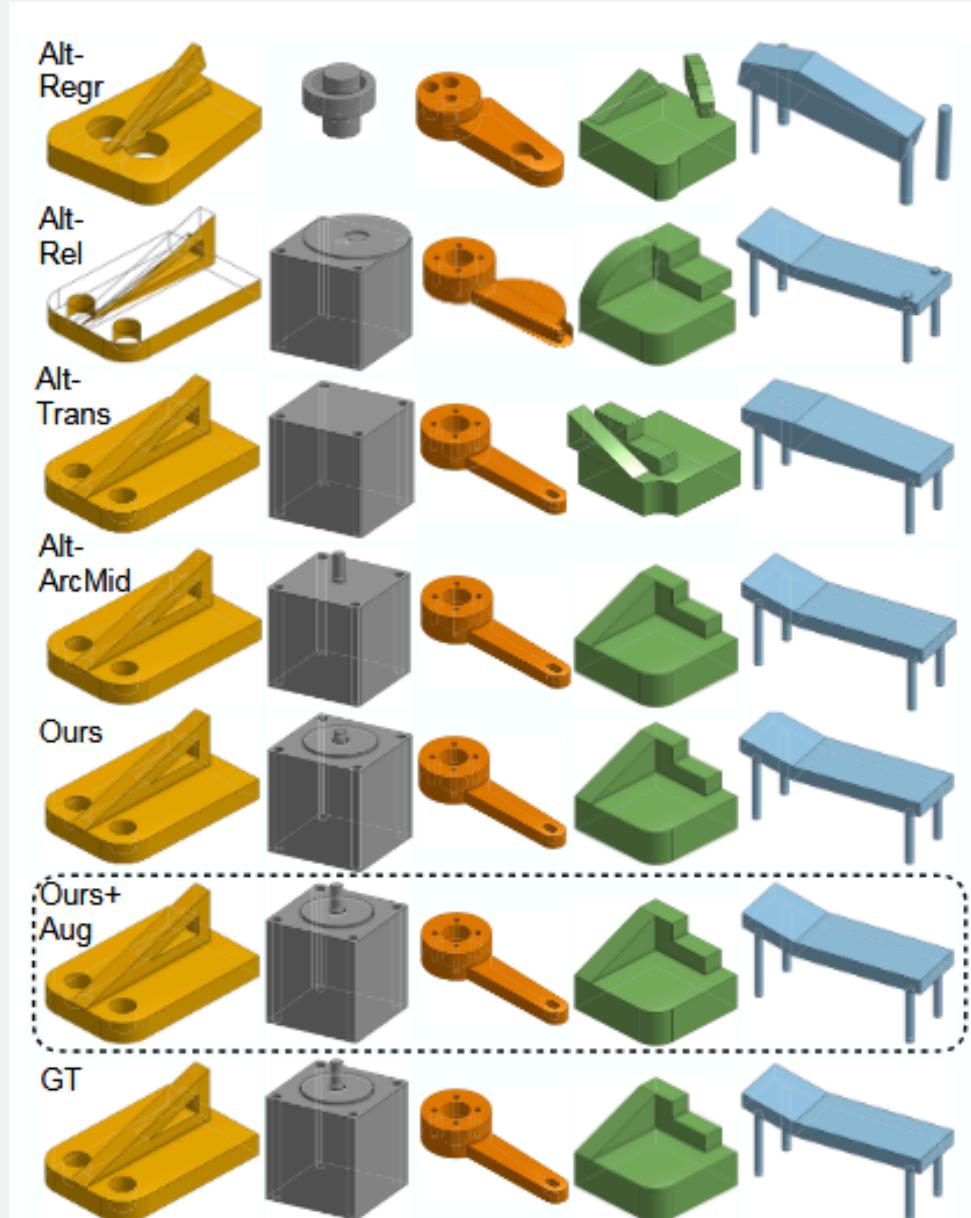
Alt-Regr은 손실 함수의 표준 평균 제곱 오차를 사용하여 CAD 명령의 모든 매개변수를 회귀합니다.

Ours+Aug는 제안하는 솔루션과 동일한 데이터 표현 및 학습 목표를 사용하지만, 무작위로 구성된 CAD 명령 시퀀스를 포함하여 학습 데이터 세트를 확장합니다(단, 확장된 시퀀스가 유효하지 않은 CAD 시퀀스인 경우도 있음).

# 본론

## 연구 방법

### 문제점



Alt-Rel이 우리 모델보다 훨씬 높은 매개변수 정확도(ACCparam)를 가지고 있지만, 상대적으로 높은 CD 점수와 때로는 유효하지 않은 토폴로지를 갖는다는 것입니다. 예를 들어, 그림 4의 두 번째 행에 있는 노란색 모델은 서로 교차하는 두 개의 삼각형 루프를 가지고 있어 유효하지 않은 토폴로지를 생성합니다. 이는 예측된 곡선 위치의 오류로 인해 발생합니다. Alt-Rel에서 곡선 위치는 이전 곡선을 기준으로 지정되므로 오류가 루프를 따라 누적됩니다. 마지막으로, 연속 매개변수를 양자화하지 않는 Alt-Regr은 평행 및 수직 모서리와 같은 곡선의 기하학적 관계를 깨뜨릴 수 있는 더 큰 오류를 겪습니다

# 본론

## 연구 방법

### Cross-dataset generalization



Figure 11. Failure examples in shape autoencoding. Top: our reconstructed CAD outputs. Bottom: ground-truth CAD models.

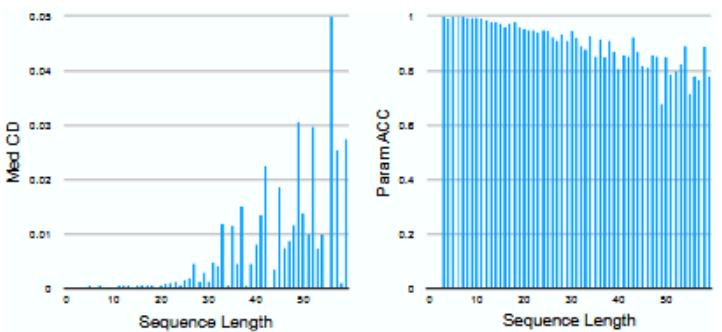


Figure 12. Quantitative metrics for shape autoencoding w.r.t. CAD sequence length. Left: median chamfer distance (the lower the better). Right: parameter accuracy (the higher the better).

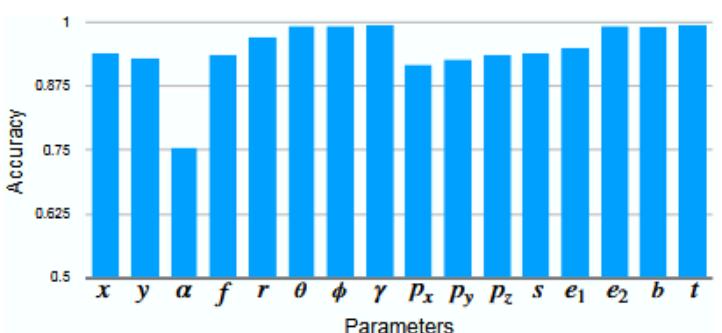


Figure 13. Accuracies for individual parameter types.

또한 자동 인코더의 일반화를 검증합니다. 생성된 데이터셋으로 학습된 자동 인코더를 [48]에서 제공된 더 작은 데이터셋에서 평가합니다. 이 데이터셋들은 서로 다른 소스에서 구축되었습니다. 저희 데이터셋은 Onshape 저장소의 모델을 기반으로 하는 반면, 다른 데이터셋은 Autodesk Fusion 360의 설계를 기반으로 제작되었습니다. 그럼에도 불구하고, 저희 네트워크는 다른 데이터셋에서도 잘 일반화되어 유사한 정량적 성능을 달성합니다

# 본론

## 연구 방법

### 결과

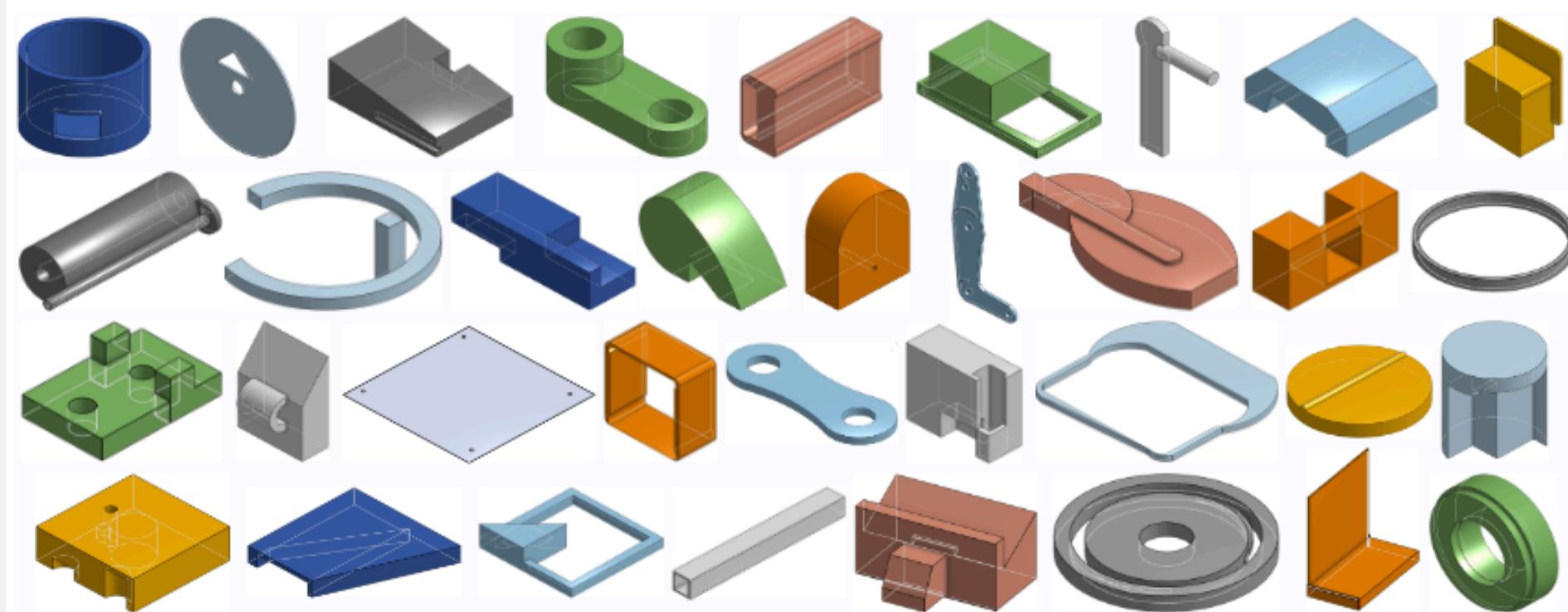


Figure 14. A gallery of our generated CAD models.



Figure 5. Randomly generated 3D shapes from our model (top) and I-GAN (bottom).

왼쪽 그림은 생성된 결과 그림입니다.

오른쪽 그림은 연구에서 만든 모델을 사용했을 때와 I-GAN 모델을 사용해서 모델을 만들었을 때, 사진 비교입니다.

CAD 설계를 위한 기준 생성 모델이 없기 때문에, 본 연구에서는 널리 연구된 포인트 클라우드 3D 형상 생성 모델인 I-GAN[6]과 본 연구의 모델을 비교하기로 했습니다.

두 생성 모델의 적용 분야가 다르기 때문에, 본 연구의 목표는 두 모델의 우월성을 입증하는 것이 아닙니다.

오히려 포인트 클라우드 생성 모델의 기준치에서도 본 연구의 모델이 유사한 형상 품질을 생성할 수 있음을 보여줍니다.

또한, 그림 5와 같이 본 연구 모델의 형상은 훨씬 더 선명한 기하학적 세부 정보를 가지며, 사용자가 쉽게 편집할 수 있습니다

# 본론

## 연구 방법

### 결과



Figure 5. Randomly generated 3D shapes from our model (top) and I-GAN (bottom).

Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 40–49, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. 1, 2, 6, 7, 8, 4

포인트 클라우드 생성 모델과의 정량적 비교를 위해 I-GAN[6]에서 사용되는 지표를 따릅니다.

이러한 지표는 두 세트의 3D 포인트 클라우드 모양, 즉 실제 값(ground-truth) 모양 세트  $S$ 와 생성된 모양 세트  $G$  간의 불일치를 측정합니다. 특히, 커버리지(COV)는  $S$ 의 모양 중  $G$ 의 모양으로 잘 근사될 수 있는 비율(%)을 측정합니다. 최소 매칭 거리(MMD)는  $S$ 와  $G$ 의 두 포인트 클라우드 사이의 최소 매칭 거리를 통해  $G$ 의 충실도를 측정합니다. 젠슨-섀넌 발산(JSD)은  $S$ 와  $G$ 의 포인트 클라우드 분포 간의 유사성을 측정하는 표준 통계적 거리입니다.

# 본론

연구 방법

결과

Method	COV↑	MMD↓	JSD↓
Ours	<b>78.13</b>	<b>1.45</b>	<b>3.76</b>
I-GAN	<b>77.73</b>	<b>1.27</b>	<b>5.02</b>

측 모델과 생성된 CAD 모델을 포인트 클라우드로 변환하고 지표를 평가합니다. 결과는 표 3에 보고되어 있으며, 이는 포인트 클라우드 지표 측면에서 본 방법이 I-GAN과 유사한 성능을 보임을 나타냅니다. 그럼에도 불구하고, CAD 모델은 매개변수 표현 덕분에 포인트 클라우드보다 훨씬 매끄러운 표면과 선명한 기하학적 특징을 가집니다.

# 결론

## 결론

CAD 설계를 위한 심층 생성 모델인 DeepCAD를 소개했습니다. 기존의 거의 모든 3D 생성 모델은 복셀, 포인트 클라우드, 메시와 같은 개별 3D 형상을 생성합니다. 본 연구는 저희가 아는 한, CAD 설계를 위한 최초의 생성 모델입니다.

Thank you

감사합니다

- | 김민성
- | 000-0000-0000
- | minsung1066@gmail.com