Boosting 3D Part Segmentation Performance with Parameter-free Lightweight Modules

Narumi Kihara [1], Namiko Kimura-Nomoto [2], Takako Okawachi [3], Guangxu Li [4,5], Norifumi Nakamura [2,6], and Tohru Kamiya [1,*]

[1] Department of Mechanical and Control Engineering, Kyushu Institute of Technology, Fukuoka, Japan
[2] Department of Oral and Maxillofacial Surgery, Kagoshima University, Kagoshima, Japan
[3] National Hospital Organization Kagoshima Medical Center, Kagoshima, Japan
[4] School of Electronics and Information Engineering, Tiangong University, Tianjin, China
[5] Eye Institute and School of Optometry, Tianjin Medical University Eye Hospital, Tianjin, China
[6] Department of Oral and Maxillofacial Surgery, Universitas Indonesia, Jawa Barat, Indonesia
Email: kihara.narumi101@mail.kyutech.jp (N.K.); nkimura@dent.kagishima-u.ac.jp (N.K.-N.); okawachi.takako.em@mail.hosp.go.jp (T.O.); liguangxu@tiangong.edu.cn (G.L.); gigi1394nakamura@gmail.com (N.N.); kamiya@cntl.kyutech.ac.jp (T.K.)
*Corresponding author

***Abstract*—Face segmentation is crucial for various facial analysis tasks. Although significant advancements have been made in deep learning-based 2D face segmentation, detailed part segmentation of 3D faces from point clouds remains largely unexplored due to annotation difficulties and high computational costs. In this paper, we propose an efficient approach for 3D facial part segmentation. Employing an efficient transformer-based model, we introduce two parameter-free lightweight modifications to improve accuracy without sacrificing computational speed: SimAM (Simple Attention Module), an attention module for salient feature enhancement, and BACE (Boundary Aware Cross Entropy), a boundary-aware loss function utilizing frozen sparse convolution with no learnable parameters. We evaluate our method on a private facial dataset that includes complex geometries, such as cleft lip deformities and various facial expressions. The experimental results demonstrate that SimAM and BACE work together to handle these complexities, leading to a statistically significant improvement. Furthermore, we validate the generalizability of the proposed modules using a dataset of 3D objects. Our method yields performance improvements for generic 3D objects when evaluated via per-sample mIoU to prevent unfair penalties from absent classes. In conclusion, integrating SimAM and BACE provides an effective and practical approach to boosting 3D part segmentation performance with minimal computational overhead.**

***Keywords*—3D point cloud, part segmentation, lightweight modules, boundary aware loss, cleft lip**

I. Introduction

Face segmentation is a technique that divides a face image into fine-grained regions, such as the nose, eyes, and mouth. Due to its wide range of applications, including facial expression recognition, head pose estimation, and facial landmark detection, it has received a great deal of attention in the field of computer vision. Because of the impressive performance of deep learning methods in numerous image processing tasks, various deep learning-driven approaches for face segmentation have emerged in recent years [1].

Training deep learning models in a supervised manner requires large-scale datasets with precise ground truth annotations. The recent public availability of numerous 2D facial image datasets has facilitated extensive research in 2D face segmentation using deep learning. On the other hand, to the best of our knowledge, the detailed part segmentation of 3D faces remains largely unexplored. This is primarily due to the high cost and difficulty of annotating facial parts in 3D data. However, since many researchers are focusing on 3D face recognition [2], 3D face segmentation is an important area that warrants further exploration.

In our previous work [3], we presented a new approach to facial part segmentation using 3D point cloud data. This approach was based on two main strategies: (1) creating a large-scale synthetic dataset with detailed pseudo-labels using 2D images, and (2) pre-training a model on this synthetic dataset, followed by fine-tuning on our small-scale private dataset. In this paper, we introduce two lightweight modifications, a parameter-free attention module and a boundary-aware loss function, to the model used in our previous work, and evaluate its performance. Although our primary focus is on 3D face segmentation, we also test our modifications on another generic 3D object dataset to validate their generalizability and effectiveness.

Our main contributions are summarized as follows:

(1) We propose a lightweight training strategy integrating parameter-free modules into PTv3 [4]. This approach effectively compensates for the baseline's inherent

Manuscript received May 7, 2026; revised May 15, 2026; accepted June 24, 2026; published November 14, 2026.

weakness in capturing fine-grained local geometries without compromising its extreme computational efficiency.

(2) We formulate BACE (Boundary Aware Cross Entropy), a parameter-free loss function utilizing a frozen sparse convolution. Unlike conventional methods, it eliminates the need for computationally expensive distance calculations or complex sub-networks.

(3) We validate the proposed framework on both a real-world clinical facial dataset and a generic 3D object dataset, demonstrating its architectural generalizability.

The rest of this paper is organized as follows: in Section II, we provide a literature review in detail, including face recognition and segmentation. Section III describes our proposed method. Section IV presents experimental results and discussion. Section V presents the conclusions.

II. Literature Review

A. 3D Face Recognition

Just as precise face segmentation is fundamental to many 2D face analysis tasks [2], it also plays a crucial role in 3D face recognition. For instance, 3D face segmentation can be used for facial landmark detection [5], face matching [6], estimation of gender, ethnicity and age [7], and facial expression recognition [8, 9]. Furthermore, our research group has proposed methods for facial symmetry analysis using 3D point cloud data [10, 11]. Our work has focused on clinical analysis for patients with cleft lips, and similar studies can be found in Refs. [12, 13]. These methods are important for improving surgical planning and assessment of cleft lip repair surgeries, which enhances the quality of life of patients. Currently, we are working on face segmentation from 3D point cloud data to create a more detailed, reliable facial analysis system.

B. Face Segmentation

In recent years, there have been significant advancements in deep learning-based 2D face segmentation, largely due to public datasets such as HELEN [14], FASSEG [15], and LaPa [16]. These datasets provide extensive collections of facial images alongside their corresponding ground truths. For instance, LaPa contains over 22,000 images, each of which is accompanied by an 11-category semantic label map. This rich information allows for highly effective face parsing based on deep learning [17, 18].

Recently, several facial 3D datasets have become publicly available, including KinectFaceDB [19] and FaceScape [20]. These datasets offer a range of facial expressions, along with supplementary information such as the positions of facial landmarks. However, unlike the 2D datasets discussed above, these datasets lack ground truth annotations of facial parts. To address this limitation, we introduced 3DDFA-V3 (3D Dense Face Alignment V3 [21]) in our previous work [3]. This method reconstructs 3D faces from 2D images. Each point in the reconstructed 3D data is automatically assigned one of eight categorical labels: “Face”, “Nose”, “Left Eyebrow”, “Right Eyebrow”, “Left Eye”, “Right Eye”, “Upper Lip” and “Lower Lip”. While these pseudo-labels may not always be perfectly accurate (e.g., the lip contours often deviate from the strict anatomical definition of the vermilion), they are highly effective as synthetic data for pre-training a deep learning model to learn global facial geometry.

C. Point Cloud Segmentation

As previously mentioned, we use point cloud representation to capture 3D facial geometry. Point clouds are commonly used to describe surface geometry and environments in scenes. They are also applied to object classification, environment mapping, and obstacle detection in robotics [22]. The most well-known point cloud segmentation datasets include S3DIS [23] and SemanticKITTI [24], which provide large-scale indoor and outdoor scenes, respectively. This highlights the fact that the mainstream focus of point cloud segmentation has been on recognizing the environment at the scene level for autonomous control, leaving relatively unexplored fine-grained tasks like deep learning-based 3D face segmentation. Furthermore, while recent advancements have explored integrating explicit boundary-prediction modules or boundary-specific decoding branches into transformer architectures to learn more distinct edges (e.g., EASE [25], PuzzleNet [26]), these modifications often increase computational overhead. In contrast, our proposed framework takes a decoupled strategy, injecting boundary awareness through a parameter-free loss function without altering the efficient backbone architecture.

III. Materials and Methods

A. Method

In clinical applications where computational resources are often limited, models must achieve high accuracy while maintaining low computational costs. Therefore, we select PTv3 (Point Transformer V3) [4] as our baseline architecture for face segmentation from 3D point clouds. As emphasized in their work [4], PTv3 has overcome the trade-off between accuracy and efficiency, achieving lower memory consumption and faster inference speeds compared to its predecessors. To justify our baseline selection, we also conduct comparative experiments using three representative alternative architectures: SPUNet (a standard voxel-based CNN) [27], Stratified Transformer (a model focused on capturing long-range dependencies) [28], and Point Transformer V2 (the predecessor to PTv3) [29].

PTv3 has achieved state-of-the-art results on many indoor and outdoor benchmarks primarily owing to its serialization-based attention mechanism and spconv-based positional encoding. However, our focus is on fine-grained part segmentation, and the serialization-based approach might struggle to capture detailed local context. Therefore, to enhance its capability to extract fine-grained local

features while maintaining efficiency, we introduce two simple lightweight modifications: SimAM (Simple Attention Module) [30] and BACE (Boundary Aware Cross Entropy). The model architecture of our proposed method is shown in Fig. 1.

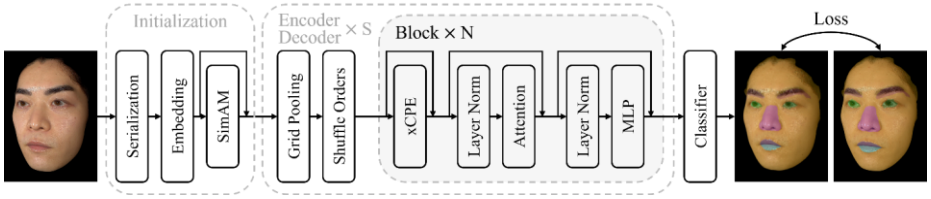


Fig. 1. Model architecture.

1) SimAM

To improve the quality of the initial feature embeddings without reducing the efficiency of PTv3, we incorporate SimAM into the initialization stage of the network. There are three reasons for this:

First, unlike conventional attention modules (e.g., SE [31] and CBAM [32]) that rely on additional learnable parameters, SimAM calculates attention weights using a fast closed-form solution of an energy function. This zero-parameter property aligns well with PTv3's goal of achieving a lightweight, highly scalable, and memory- efficient architecture.

Algorithm 1: Pseudocode of SimAM for batched point clouds

```
Function SimAM (X, B, lambda):
# Inputs:
# X: Input point cloud features of shape [N, C]
#     (N: total number of points, C: channel size)
# B: Batch indices of shape [N]
#    (e.g., [0, 0, ..., 1, 1, 1, ...])
# lambda: Regularization coefficient

# 1. Calculate the spatial mean per sample and broadcast to
each point
mu_per_sample = Scatter_Mean(X, B)
mu = mu_per_sample[B]  # Broadcast back to shape [N, C]

# 2. Calculate the squared difference from the mean
d = (X - mu)^2

# 3. Calculate the spatial variance per sample and broadcast
v_per_sample = Scatter_Mean(d, B)
v = v_per_sample[B]

# 4. Calculate attention weights
E_inv = d / (4 * (v + lambda)) + 0.5

# 5. Feature refinement
Output = X * sigmoid(E_inv)

return Output
```

Second, SimAM directly infers full 3D weights by jointly evaluating the importance across spatial and channel dimensions. This module was originally designed for 2D images with a tensor shape of (B, C, H, W), where B, C, H, and W are the batch size, the channel size, the height, and width, respectively. However, it can be efficiently adapted to batched 3D point clouds represented as a feature tensor of shape (N, C) and a corresponding batch index tensor of shape (N), where N is the total number of points. As detailed in Algorithm 1, we utilize scatter operations based on the batch indices to calculate the spatial mean and variance per sample and per channel. These statistics are then broadcasted back to the (N, C) shape. This tensor-level adaptation mathematically preserves the original spatial and channel attention mechanisms in the unstructured point cloud domain.

Finally, SimAM draws inspiration from surround suppression in mammalian visual neuroscience, where highly active neurons suppress the activity of surrounding neurons to highlight salient stimuli. SimAM minimizes the energy function to measure the linear separability of each point from its local neighborhood, assigning higher importance to points that exhibit distinctive patterns. With this module in the initialization stage, the network can effectively emphasize critical geometric structures early in the pipeline.

2) Boundary aware cross entropy

In 3D point cloud segmentation, boundary ambiguity between objects or parts remains a critical challenge. It has been pointed out that features are highly susceptible to mixing near boundaries due to the expansion of the receptive field, leading to a significant degradation in segmentation accuracy [33, 34].

To address this issue, most existing studies have adopted approaches incorporating specialized architectures or sub-networks dedicated to boundary detection [35]. In the field of 2D face parsing, for example, methods have been proposed that introduce an auxiliary edge estimation branch during training and decouple it during testing to reduce computational overhead [18]. However, while these methods successfully reduce the burden during testing, they require the additional learnable parameters and complex sub-networks during training. This leads to an undesirable increase in memory consumption and training computational cost, particularly for 3D point cloud data.

In contrast, approaches that improve boundary prediction accuracy by optimizing loss functions, without altering the network architecture have garnered significant attention [36, 37]. These methods are advantageous in terms of simplicity.

In this study, we propose using BACE for 3D point cloud segmentation, which minimizes the computational cost even during training. Our approach eliminates the need for any complex, dedicated mechanisms or computationally heavy distance calculations. Instead, our

loss function introduces a frozen 3 × 3 × 3 sparse convolution. Therefore, it can be seamlessly integrated into any point cloud networks (including PTv3) that utilize voxel-based sparse convolutions to improve boundary representations.

We provide detailed formulation of our BACE. Let $\mathcal{P} = \{p_i\}_{i=1}^{N}$ be the input point cloud and $y_i \in \{1, \dots, C\}$ be the corresponding ground truth labels. Each point p_i is quantized into a voxel coordinate v_i based on a predefined grid size (generally set in a config file as a hyperparameter). Each voxel is then represented by a one-hot vector of the ground truth label, denoted as $h_v \in \{0,1\}^C$. To dynamically capture the class distribution within the local neighborhood $\mathcal{N}(v)$ of the voxel space, we apply a submanifold sparse convolution with a kernel size 3 × 3 × 3. This specific kernel size was chosen because it defines the minimal neighborhood required to efficiently capture local boundary transitions with minimal computational cost.

Instead of using learnable parameters, we fix the convolutional weights to an identity matrix, which ensures channel independence. Consequently, the output feature $o_v \in \mathbb{R}^C$ counts the number of points for each class within the neighborhood:

$$o_v = \sum_{u \in \mathcal{N}(v)} h_u \tag{1}$$

Next, we determine whether each point p_i is located on a boundary. A point is considered to lie on a boundary if there are more than two distinct semantic classes within its local neighborhood. This can be formulated by simply counting the number of dimensions in the aggregated feature o_v, where $\mathbb{I}(\cdot)$ is an indicator function that equals 1 if the condition is true and 0 otherwise. We then obtain a boundary indicator b_i.

$$b_i = \mathbb{I}\left(\sum_{c=1}^{C} \mathbb{I}\left(o_{v,c} > 0\right) > 1\right) \tag{2}$$

Furthermore, to selectively emphasize the boundaries of specific parts (named CoI (Class of Interest) in our code), we compute a boundary mask m_i using a target class set C_{CoI} . If the CoI is not givento the loss function, all boundaries are used.

$$m_i = b_i \cdot \mathbb{I}(y_i \in C_{CoI}) \tag{3}$$

Based on this mask, we assign a dynamic penalty weight w_i to each point, where λ is a hyperparameter that controls the contributions of boundaries.

$$w_i = \begin{cases} \lambda & \text{if } m_i = 1 \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

The final loss function L_{BACE} is obtained by performing an element-wise multiplication of the standard cross-entropy loss $L_{CE}(p_i)$ with the dynamic weight w_i:

$$L_{BACE} = \frac{1}{N} \sum_{i=1}^{N} w_i \cdot L_{CE}(p_i) \tag{5}$$

In our implementation, we empirically selected $\lambda = 2$. This exploration remains a topic for future work.

B. Dataset

1) Face segmentation

As in our previous work [3], we collected the CelebA-HQ dataset [38, 39] and used the first 24,000 images. The 3DDFA-V3 was applied to these 2D images to generate facial 3D point clouds and pseudo-labels, creating a large-scale synthetic dataset. This collection of facial 3D point clouds is referred to as CelebA-3D.

Additionally, we collected a small-scale private dataset [3]. These point clouds were captured at Kagoshima University in Japan using a VECTRA®H1 manufactured by Canfield Scientific. The breakdown of this dataset is shown in Table I. For Table I(a), participants were asked to sit on a chair and relax. For Table I(b) and (c), in addition to the relaxed expression, participants were asked to open their mouths to analyze their facial expressions. We note that all participants with cleft lips had already undergone the first surgery.

TABLE I. DETAILS OF OUR FACIAL DATASET

Group	Cleft lip	Subjects	Total data
(a)	w/	15	15
(b)	w/	8	21
(c)	w/o	5	19

Although only pseudo-labels were used for CelebA-3D, we manually created ground truths for the upper and lower lips under the strict guidance of clinical specialists. To eliminate inter-observer variability and ensure consistency, a single annotator performed all labeling based on predefined anatomical criteria, specifically defining the lip region as the vermilion area bounded by the inner edge of the white roll. Subsequently, the clinical specialists reviewed the annotated 3D point clouds to validate their anatomical accuracy. We then replaced the pseudo-labels with the ground truths for the private dataset [3]. This is because we intend to apply our method to patients with cleft lips and associated lip deformities.

2) Object segmentation

Although our main focus is on face segmentation, we test our modifications on PartNeXt [40]. This dataset contains over 23,000 high-quality, textured 3D models across 50 categories. For our evaluation, we selected 6 categories (Axe, Bed, Bottle, Chair, Chandelier, Sofa) that originally contained more than 1000 samples to ensure robust training and evaluation.

We applied uniform surface sampling based on face areas to the provided GLB mesh data, extracting 32,768 points per model along with their corresponding color and normal information. To avoid the primitive splitting and index shuffling issues that are inherent in mesh loading libraries, we introduced a forced alignment method that utilizes the ground truth face count (mesh_face_num) from the annotation data. This method ensures strictly noise-free label assignment. For more stable training, we assigned an ignore label (−1) to the “Other” class and excluded it from the loss computation. Finally, we randomly partitioned the

data within each category into training and testing sets using a standard 9:1 split ratio.

Note that we excluded the "Table" category from our experiment, which contained more than 1000 samples. 3D table models frequently contain geometrically degenerate structures, such as flat planar meshes for tabletops and extremely thin structures for legs. During uniform surface sampling, these extreme aspect ratios lead to severe point density imbalances. These inherent geometric artifacts cause severe numerical instability in local feature aggregation, making the category unsuitable for stable evaluation of part segmentation methods.

C. Implementation Details

Each experiment was carried out on a single GPU, NVIDIA GeForce RTX 3060Ti (8GB of VRAM) for facial datasets and NVIDIA RTX PRO 4000 Blackwell (24GB of VRAM) for PartNeXt. The hyperparameters for PTv3 largely followed the original configurations. The details are shown in Table II. For the alternative models (i.e., SPUNet, ST (Stratified Transformer), and PTv2 (Point Transformer V2)), we conducted experiments on both the CelebA-3D and the private datasets. While the criteria, grid size, batch size, and epochs were kept consistent with those listed in Table II, the remaining parameters, such as model structures, optimizers, and schedulers, were set to their default values as implemented in the open-source Pointcept codebase [41].

TABLE II. MODEL SETTINGS

Config.	CelebA-3D (Pre-train)	Private dataset (Fine-tune)	PartNeXt
Optimizer	AdamW		
Scheduler	Cosine		
Learning rate	2×10^{-3}		
Block lr scaler	1×10^{-1}		
Weight decay	5×10^{-3}		
Criteria	1 × Lovasz and 1 × (Cross Entropy or BACE)		
Grid size	0.02	0.02	0.01
Batch size	16	4	16
Epochs	5	200	150
Blocks	[2, 2, 6, 2]		
CoI	None	Upper/Lower lips	None

Optimizing the grid size remains an important direction for future work, because preserving local geometric information is crucial for fine-grained part segmentation tasks. In addition, while our current parameters (e.g., kernel size and λ for the BACE loss function) were selected based on theoretical efficiency and empirical observations, it is crucial to conduct a comprehensive ablation study to optimize these hyperparameters.

IV. RESULT AND DISCUSSION

A. Evaluation Protocol

1) Evaluation metric

We used IoU (Intersection over Union) as an evaluation metric. Let N be the total number of samples in the test dataset. The mIoU_i for the i-th sample is calculated as follows:

$$\mathrm{mIoU}_i = \frac{1}{|C_i|}\sum_{c \in C_i} \mathrm{IoU}_{i,c} \tag{6}$$

The per-sample mIoU for the entire dataset is defined as the average mIoU across all samples:

$$\text{per-sample mIoU} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{mIoU}_i \tag{7}$$

where C_i denotes the set of part classes actually present in the ground truth of the i-th sample. We used this mIoU computation because the presence of part classes varies depending on the object in the PartNeXt dataset (e.g., the class "ladder" only exists in specific beds). This formulation prevents the segmentation performance from being penalized by absent part classes, which would otherwise unavoidably yield an IoU of 0. In addition, to evaluate the segmentation quality at boundary regions, we adopted BIoU (Boundary IoU). The boundaries for this evaluation were extracted based on the same voxel-based approach as our BACE loss function. The IoU was then calculated within these regions.

We employed the Wilcoxon signed-rank test [42] with a significance level of 0.05 to statistically evaluate the performance between the proposed method and the base model. Since differences in IoU do not necessarily follow a normal distribution, parametric methods like the paired *t*-test are not appropriate [43]. As a non-parametric method, this test ranks the absolute differences in scores between methods for each sample and determines significance based on the sum of these ranks. Consequently, it provides a robust evaluation of consistent accuracy improvements at the per-sample level. This approach ensures that the results are not heavily influenced by extreme outliers from exceptionally difficult or easy samples.

2) Face segmentation

Each model was trained in two stages: pre-training on CelebA-3D and fine-tuning on our private dataset. The final accuracy of the fine-tuning stage was calculated by leave-one-subject-out cross-validation. For our implementation, we selected one participant from Table I(b) or I(c) for testing (i.e., all facial expressions of that specific participant were used for testing) and used the remaining participants for training. This subject-level split ensures that no facial identity leakage occurs between the training and testing sets. Participants from Table I(a) were only used for training because our goal was to confirm the segmentation performance for multiple facial expressions. Although the model outputs the 8-class semantic segmentation maps for each face, we focused only on the upper and lower lips using manually created ground truths.

3) Object segmentation

As previously described, the data was randomly partitioned into training and testing sets. The model was trained from scratch and tested on each category.

B. Result

1) Face segmentation

Tables III and IV show the standard mIoU and BIoU results on our facial dataset, respectively. The proposed

modules improved the segmentation accuracy; particularly in the Lower Lip segmentation, the model PTv3+BACE achieved 74.8% ($p = 0.031$), and the model PTv3+Both achieved 75.3% ($p = 0.029$), demonstrating statistically significant improvements over the baseline. Visual comparisons of the segmentation results are shown in Fig. 2.

TABLE III. RESULTS ON OUR FACIAL DATASET (IOU)

Model	Setting	Upper lip	Lower lip
SPUNet	Baseline	69.8 (0.0049)	70.4 (0.0035)
	+PT	71.9	72.8
	+PT&BACE	72.2 (0.60)	74.3 (0.064)
ST	Baseline	64.9 (8.2×10^{-8})	66.1 (4.3×10^{-7})
	+PT	71.1	74.4
	+PT&BACE	71.2 (0.97)	74.6 (0.66)
PTv2	Baseline	71.2 (0.15)	71.0 (0.012)
	+PT	72.3	74.1
	+PT&BACE	72.0 (0.40)	74.5 (0.72)
PTv3	Baseline	67.5 (1.4×10^{-4})	69.1 (1.2×10^{-5})
	+PT	71.4	73.7
	+PT&SimAM	71.2 (0.86)	74.8 (0.10)
	+PT&BACE	71.7 (0.43)	74.8 (0.031)
	+All	71.9 (0.44)	75.3 (0.029)

The models "+PT" were pre-trained on the CelebA-3D dataset. The *p*-values were calculated relative to their respective pre-trained base models.

2) Object segmentation

Tables V and VI show the standard mIoU and BIoU results on the PartNeXt dataset, respectively. Notable and statistically significant improvements were observed in the Axe and Bed categories. Specifically for Bed, the accuracy improved from the baseline (79.7%) to 85.7% ($p = 1.1 \times 10^{-12}$) with BACE and 85.0% ($p = 3.6 \times 10^{-13}$) with Both. On the other hand, in the Bottle and Chandelier categories, no notable improvement was observed with our modifications, and the accuracy even decreased with SimAM. Visual comparisons are shown in Fig. 3.

TABLE IV. RESULTS ON OUR FACIAL DATASET (BOUNDARY IOU)

Model	Setting	Upper lip	Lower lip
SPUNet	Baseline	77.7 (0.58)	78.8 (0.26)
	+PT	78.4	80.4
	+PT&BACE	80.3 (1.8×10^{-3})	83.1 (8.3×10^{-5})
ST	Baseline	72.8 (7.5×10^{-6})	71.6 (2.7×10^{-6})
	+PT	79.8	81.2
	+PT&BACE	82.0 (1.6×10^{-4})	82.9 (4.6×10^{-3})
PTv2	Baseline	79.7 (0.70)	78.1 (0.060)
	+PT	80.3	81.1
	+PT+BACE	81.7 (2.9×10^{-3})	83.1 (1.7×10^{-3})
PTv3	Baseline	74.8 (9.0×10^{-3})	74.1 (6.1×10^{-3})
	+PT	78.6	78.3
	+PT&SimAM	80.5 (1.1×10^{-4})	81.4 (6.3×10^{-5})
	+PT&BACE	78.9 (0.81)	81.4 (5.1×10^{-4})
	+All	81.2 (6.2×10^{-5})	83.5 (3.5×10^{-8})

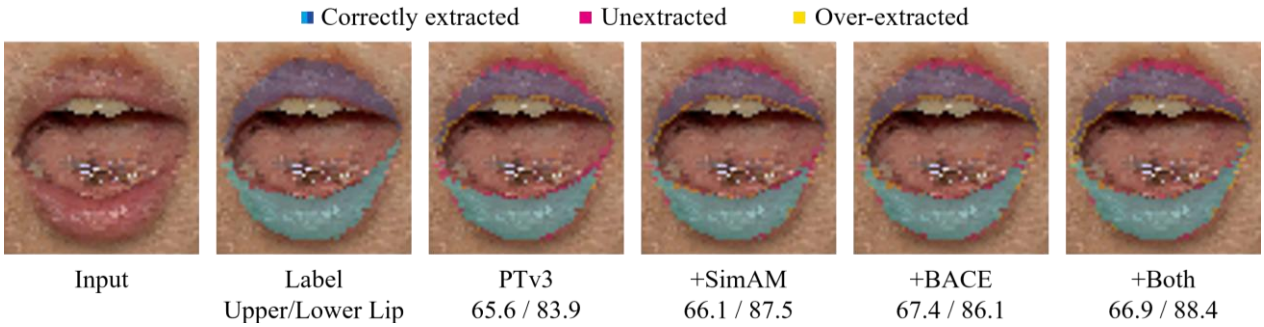


Fig. 2. Visual comparisons (our facial dataset).

TABLE V. RESULTS ON PARTNEXT DATASET (IOU)

Category	PTv3	+SimAM	+BACE	+Both
Axe	86.2	87.6 (0.00027)	89.6 (1.4×10^{-9})	88.9 (6.3×10^{-9})
Bed	79.7	84.1 (2.3×10^{-11})	85.7 (1.1×10^{-12})	85.0 (3.6×10^{-13})
Bottle	72.1	70.5 (0.0048)	73.3 (0.074)	72.7 (0.34)
Chair	81.9	82.6 (0.53)	82.1 (0.029)	82.2 (0.14)
Chandelier	60.4	56.7 (0.00097)	60.3 (0.93)	60.8 (0.077)
Sofa	86.6	86.8 (0.10)	86.5 (0.72)	86.5 (0.75)

TABLE VI. RESULTS ON PARTNEXT DATASET (BOUNDARY IOU)

Category	PTv3	+SimAM	+BACE	+Both
Axe	41.1	49.5 (2.1×10^{-7})	59.8 (2.5×10^{-10})	59.8 (2.6×10^{-10})
Bed	61.6	68.0 (6.9×10^{-13})	76.7 (7.3×10^{-18})	75.4 (2.4×10^{-16})
Bottle	35.0	35.1 (0.86)	35.9 (0.96)	37.5 (0.13)
Chair	58.5	59.2 (0.41)	61.1 (1.2×10^{-11})	60.6 (6.0×10^{-8})
Chandelier	42.4	39.4 (1.4×10^{-3})	44.2 (0.011)	45.6 (4.3×10^{-5})
Sofa	77.5	77.4 (0.045)	78.2 (2.8×10^{-9})	78.0 (9.7×10^{-8})

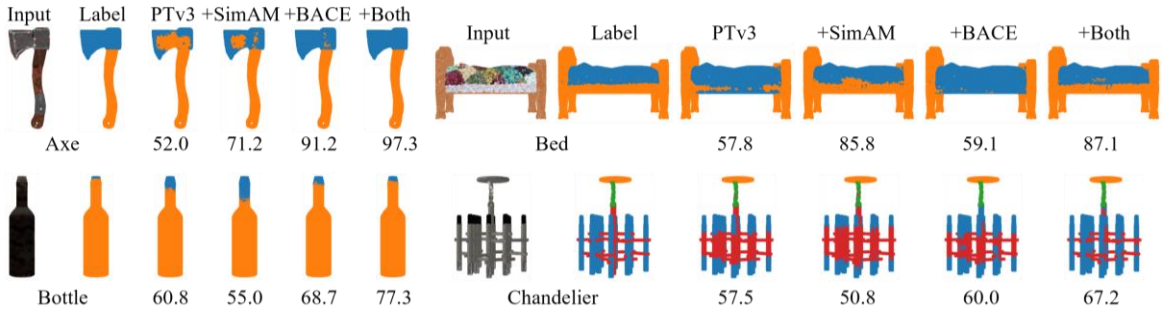


Fig. 3. Visual comparisons (PartNeXt dataset).

3) *Computational cost*

To quantify the computational overhead of our proposed modules and compare the efficiency across different model architectures, we measured the parameter count, GPU memory consumption (allocated and reserved), and latency during both training and inference. The measurements were conducted using the CelebA-3D dataset. As shown in Tables VII and VIII, integrating SimAM and BACE into PTv3 introduces no additional learnable parameters and no increase in memory consumption. The latency increase is marginal (+10.3% in total for training). Furthermore, compared to alternative transformer-based models (PTv2 and ST), PTv3 demonstrates advantages in memory efficiency and inference speed.

TABLE VII. COMPARISONS OF PARAMETER COUNT AND PROCESSING LATENCY

Model	Parameters (M)	Training latency (ms)	Inference latency (ms)
SPUNet	39.2	106	50
ST	18.8	251	183
PTv2	3.9	162	65
PTv3	46.2	165	78
+BACE	46.2	168	80
+SimAM	46.2	179	82

Measured on the CelebA-3D Dataset.

TABLE VIII. COMPARISONS OF GPU PEAK MEMORY CONSUMPTION

Model	Training		Inference	
	Allocated (G)	Reserved (G)	Allocated (G)	Reserved (G)
SPUNet	0.49	0.61	0.24	0.32
ST	0.51	1.03	0.25	0.86
PTv2	1.34	2.37	0.25	0.63
PTv3	0.55	0.61	0.23	0.26
+BACE	0.55	0.61	0.23	0.26
+SimAM	0.55	0.62	0.23	0.26

Measured on the CelebA-3D Dataset.

C. *Discussion*

1) *Face segmentation*

The performance improvements in our facial dataset can be attributed to the complementary roles of SimAM and BACE in handling the complex geometry of the facial region, particularly the lips.

First, the effectiveness of the BACE module was evident. It effectively addressed the boundary ambiguity between lips and skin, even with facial expression variations (e.g., mouth opening) in our dataset. This approach significantly improved the boundary prediction without introducing high memory consumption or computational costs. Consequently, the Lower Lip segmentation accuracy showed a statistically significant improvement from 73.7% to 74.8% ($p = 0.031$).

Second, while SimAM alone improved the Lower Lip accuracy to 74.8%, it did not reach statistical significance ($p = 0.10$). Human faces are generally smooth, which may limit the standalone potential of SimAM, a module designed to emphasize spatially distinct and salient features. However, when SimAM was combined with BACE, a synergistic effect emerged, achieving the highest accuracy and the most notable result for the Lower Lip (75.3%, $p = 0.029$). This synergy suggests that SimAM enhances the globally salient structural features in the early initialization stage of the network, while BACE explicitly supervises the fine-grained boundary separation during the loss computation. This dual enhancement ultimately led to the most robust segmentation performance for complex facial parts. Particularly for the Upper Lip, while the improvement in standard mIoU was marginal and lacked statistical significance due to its extreme geometric variations, the BIoU metric revealed a statistically significant improvement. This demonstrates that our proposed modules refined the critical boundary regions even around complex structures like the cupid's bow.

As shown in Fig. 2, these qualitative results support the synergistic effect. In the baseline, misclassifications that are either over-extracted (yellow) or unextracted (magenta) frequently occur around the lip regions. Adding SimAM or BACE alone mitigates these errors to some extent, but combining both modules effectively refines the detailed boundaries. Consequently, it achieves the best performance, particularly for the lower lip.

2) *Object segmentation*

The results indicate that the effectiveness of the proposed modules depends on the shape characteristics of the objects. For the Axe category (an object with a relatively simple shape) and the Bed category (an object with large parts such as frames and sleeping areas), the boundary enhancement effect of BACE and the reinforcement of global salient features by SimAM worked effectively. As demonstrated in Fig. 3, these results provide evidence in support of this observation. For the Axe category, BACE eliminated the misclassification between the head and handle, producing clearer

boundaries. For the Bed category, while SimAM alone roughly captured the global structures of the frame and sleeping area, the boundaries remained somewhat coarse. The combination of both modules effectively refined these boundaries, visually demonstrating a synergistic effect.

Conversely, the performance degradation observed in the Bottle and Chandelier categories with SimAM alone can be explained by their extreme geometric properties. Fundamentally, SimAM is designed to enhance spatially and channel-wise salient features. In the Bottle category, which is characterized by smooth surfaces with very few sharp geometric transitions, this lack of salient features causes SimAM to struggle in assigning meaningful attention weights. On the other hand, the Chandelier category consists of highly complex and scattered structures. Applying SimAM to such complex geometries leads to the over-amplification of local geometric noise and minor details. Consequently, this reduces the accuracy compared to the baseline.

However, a detailed qualitative analysis shows that combing SimAM with BACE (+Both) can be effective for specific complex cases, as demonstrated by the bottom two rows of Fig. 3. For the Chandelier sample, the boundary penalty of BACE prevented the geometric collapse, turning SimAM's feature enhancement into a positive contribution and achieving the highest accuracy. Similarly, for the Bottle sample, the combination of both modules effectively captured the small cap region that the baseline model failed to segment. These visual improvements are also quantitatively supported by the BIoU metric in Table VI. For instance, in the geometrically complex Chandelier category, the combination of both modules demonstrated a statistically significant gain in BIoU (from 42.4% to 45.6%, $p = 4.3 \times 10^{-5}$). This boundary-level improvement validates our hypothesis that SimAM's global feature enhancement and BACE's local boundary supervision work complementarily.

3) Architecture selection and computational efficiency

Our comprehensive comparisons with alternative backbones further justify the selection of PTv3 and demonstrate the effectiveness of our lightweight modules.

First, regarding the comparison with the CNN-based SPUNet, PTv3 exhibited a larger performance gain from large-scale pre-training (+4.6% vs. +2.4% in the lower lip mIoU). While SPUNet performs relatively well when trained from scratch due to its strong inductive bias, transformer-based models demonstrate superior scalability and data-hungry characteristics. This indicates that PTv3 possesses a higher potential to benefit from our future dataset expansion.

Second, compared to other transformer-based models, PTv2 and ST achieved high accuracy but suffered from severe computational bottlenecks. As shown in Tables VII and VIII, ST exhibited higher latency, and PTv2 consumed approximately 2.4 times more peak memory than PTv3 due to its complex positional encoding and kNN (k-Nearest Neighbor) operations. PTv3 prioritizes simplicity and scalability, which might compromise its ability to capture local boundaries compared to PTv2. However, our proposed modules effectively compensate for this geometric weakness without increasing memory usage.

Ultimately, the combination of PTv3 with both modules (PTv3+Both) achieved the highest accuracy (75.3% for the lower lip), surpassing the heavier models. This shows that our strategy of equipping a fast and memory-efficient model with parameter-free local enhancement modules offers the optimal balance between clinical practicality and segmentation accuracy.

D. Limitations and Future Work

Despite the promising results, our study has several limitations that provide important directions for future work.

First, our framework relies on voxelization and is potentially sensitive to variations in sparse sampling density and real-world acquisition noise (e.g., missing regions). Although we mitigated this by employing point cloud data augmentations during training (e.g., random dropout, random jitter, and sphere cropping), we have not yet conducted a quantitative robustness analysis under controlled noisy conditions. Comprehensive sensitivity analysis regarding grid size dependency and data perturbations will be addressed in future studies.

Second, performance variations exist across complex geometries. For instance, while improvements for the lower lip were statistically significant, the standard mIoU gains for the upper lip remained marginal. This is fundamentally due to the extreme geometric variations caused by cleft deformities and complex local contour of the cupid's bow.

Finally, although pre-training on the large-scale CelebA-3D dataset mitigates the risk of overfitting, the relatively small size of our private facial dataset remains a limitation. Additionally, investigating the model's sensitivity to pseudo-label noise during pre-training is necessary.

In future work, we will explore novel 3D point cloud data augmentation techniques to cover severe anatomical variations. Furthermore, we intend to deepen our collaboration with clinical partners to continuously expand our dataset. This expansion will encompass a broader range of cases spanning various age groups and degrees of cleft lip severity. This dataset expansion will be crucial to further validating the reliability, robustness, and generalization capabilities of our proposed method. Alongside this expansion, we also plan to incorporate additional multi-faceted evaluation metrics (e.g., Dice score) to provide a more comprehensive assessment of the segmentation quality. Finally, we plan to integrate our segmentation framework into downstream clinical workflows. The fine-grained boundary refinement achieved by our proposed method provides a reliable foundation for quantitative facial symmetry analysis of patients with cleft lips, which utilizes 3D non-rigid registration of the extracted lip regions. Ultimately, this will supply plastic surgeons with objective baselines for pre-operative surgical planning and post-operative evaluations.

V. Conclusion

In this paper, we investigated 3D point cloud part segmentation using an efficient transformer-based model. To enhance the accuracy of the segmentation process without sacrificing computational efficiency, we introduced two parameter-free, lightweight modifications. SimAM is an attention module that enhances salient features. BACE is a boundary-aware loss function that uses frozen sparse convolution.

Evaluations on our private dataset showed that their synergistic combination resolved boundary ambiguity, significantly improving Lower Lip segmentation. Furthermore, validations on the PartNeXt dataset confirmed performance improvements for generic 3D objects.

In conclusion, integrating SimAM and BACE boosts 3D part segmentation performance with minimal overhead. Future work includes optimizing BACE's hyperparameters (e.g., penalty weight λ) and grid size, and integrating our framework into clinical assessment systems for patients with cleft lips.

Data and Code Availability

To support reproducibility, the source code of the modules (SimAM and BACE), the model configuration files, the scripts for generating the CelebA-3D dataset, and the preprocessing scripts for the PartNeXt dataset will be made publicly available. The private facial dataset cannot be shared due to strict privacy and ethical restrictions.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Narumi Kihara conducted the research and wrote the paper. Namiko Kimura-Nomoto and Takako Okawachi collected the facial data, verified the quality of the ground truth, and examined the research from a medical perspective. Guangxu Li, Norifumi Nakamura, and Tohru Kamiya acted as academic supervisors and reviewed the research in terms of engineering, computer vision, and medical imaging. All authors have approved the final version.

Funding

This work was supported by JST SPRING, Japan Grant Number JPMJSP2154.

References

[1] K. Khan, R. U. Khan, K. Ahmad *et al*., "Face Segmentation: A Journey from Classical to Deep Learning Paradigm, Approaches, Trends, and Directions," *IEEE Access*, vol. 8, pp. 58683–58699, 2020.

[2] S. Zhou and S. Xiao, "3D Face Recognition: A Survey," *Human-centric Computing and Information Sciences*, vol. 8, no. 35, 2018.

[3] N. Kihara, N. Kimura-Nomoto, T. Okawachi *et al*., "A Novel Deep Learning-based Approach for Facial Part Segmentation from 3D Point Cloud Data," in *Proc. International Conference on Computing and Pattern Recognition*, 2026, pp. 137–145.

[4] X. Wu, L. Jiang, P. Wang *et al*., "Point Transformer V3: Simpler, Faster, Stronger," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4840–4851.

[5] S. Liang, J. Wu, S. M. Weinberg *et al*., "Improved Detection of Landmarks on 3D Human Face Data," in *Proc. IEEE Engineering in Medicine and Biology Society*, 2013, pp. 6482–6485.

[6] Z. You, T. Yang, and M. Jin, "Multi-channel Deep 3D Face Recognition," arXiv Preprint, arXiv:2009.14743, 2020. https://arxiv.org/abs/2009.14743

[7] B. Xia, B. B. Amor, and M. Daoudi, "Joint Gender, Ethnicity and Age Estimation from 3D Faces: An Experimental Illustration of Their Correlations," *Image and Vision Computing*, vol. 64, pp. 90–102, 2017.

[8] R. Ramya, K. Mala, and S. S. Nidhyananthan, "3D Facial Expression Recognition Using Multi-channel Deep Learning Framework," *Circuits, Systems, and Signal Processing*, vol. 39, pp. 789–804, 2020.

[9] G. Sandbach, S. Zafeiriou, M. Pantic *et al*., "Recognition of 3D Facial Expression Dynamics," *Image and Vision Computing*, vol. 30, no. 10, pp. 762–773, 2012.

[10] D. Hosoki, T. Kamiya, N. Kimura-Nomoto *et al*., "Symmetric Plane Detection and Symmetry Analysis from A 3D Point Cloud Data of Face," in *Proc. International Conference on Control, Automation and Systems*, 2020, pp. 402–406.

[11] N. Kihara, N. Kimura-Nomoto, T. Okawachi *et al*., "Facial Symmetry Analysis for Cleft Lip Patients from 4D Point Cloud Data," in *Proc. International Conference on Control, Automation and Systems*, 2023, pp. 1502–1505.

[12] D. Al-Rudainy, X. Ju, S. Stanton *et al*., "Assessment of Regional Asymmetry of the Face Before and After Surgical Correction of Unilateral Cleft Lip," *Journal of Cranio-Maxillofacial Surgery*, vol. 46, no. 6, pp. 974–978, 2018.

[13] S. Gattani, X. Ju, T. Gillgrass *et al*., "An Innovative Assessment of the Dynamics of Facial Movements in Surgically Managed Unilateral Cleft Lip and Palate Using 4D Imaging," *The Cleft Palate-craniofacial Journal*, vol. 57, pp. 1125–1133, 2020.

[14] B. M. Smith, L. Zhang, J. Brandt *et al*., "Exemplar-based Face Parsing," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3484–3491.

[15] K. Khan, M. Mauro, P. Migliorati *et al*., "Head pose estimation through multiclass face segmentation," in *Proc. IEEE International Conference on Multimedia and Expo*, 2017, pp. 175–180.

[16] Y. Liu, H. Shi, H. Shen *et al*., "A New Dataset and Boundary-Attention Semantic Segmentation for Face Parsing," in *Proc. AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, 2020, pp. 11637–11644.

[17] G. Te, Y. Liu, W. Hu *et al*., "Edge-aware Graph Representation Learning and Reasoning for Face Parsing," in *Proc. Computer Vision—ECCV 2020*, 2020, pp. 258–274.

[18] Q. Zheng, J. Deng, Z. Zhu *et al*., "Decoupled Multi-task Learning with Cyclical Self-Regulation for Face Parsing," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4156–4165.

[19] R. Min, N. Kose, and J. Dugelay, "KinectFaceDB: A Kinect Database for Face Recognition," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 11, pp. 1534–1548, 2014.

[20] H. Zhu, H. Yang, L. Guo *et al*., "FaceScape: 3D Facial Dataset and Benchmark for Single-view 3D Face Reconstruction," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. 45, no. 12, 2020, pp. 14528–14545.

[21] Z. Wang, X. Zhu, T. Zhang *et al*., "3D Face Reconstruction with the Geometric Guidance of Facial Part Segmentation," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 1672–1682.

[22] Y. He, H. Yu, X. Liu *et al*., "Deep Learning Based 3D Segmentation in Computer Vision: A Survey," *Information Fusion*, vol. 115, 102722, 2025.

[23] I. Armeni, O. Sener, A. R. Zamir *et al*., "3D Semantic Parsing of Large-scale Indoor Spaces," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1534–1543.

[24] J. Behley, M. Garbade, A. Milioto *et al*., "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 9297–9307.

[25] W. Roh, H. Jung, G. Nam *et al*., "Edge-aware 3D Instance Segmentation Network with Intelligent Semantic Prior," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20644–20653.
[26] H.Y. Liu, J.W. Guo, H.Y. Jiang *et al*., "PuzzleNet: Boundary-aware Feature Matching for Non-overlapping 3D Point Clouds Assembly," *Journal of Computer Science and Technology*, vol. 38, pp. 492–509, 2023.
[27] SpConv Contributors. (2022). SpConv: Spatially Sparse Convolution Library. [Online]. Available: https://github.com/traveller59/spconv
[28] X. Lai, J. Liu, L. Jiang *et al*., "Stratified Transformer for 3D Point Cloud Segmentation," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8500–8509.
[29] X. Wu, Y. Lao, L. Jiang *et al*., "Point Transformer V2: Grouped Vector Attention and Partition-based Pooling," in *Proc. International Conference on Neural Information Processing Systems*, 2022, pp. 33330–33342.
[30] L. Yang, R. Zhang, L. Li *et al*., "SimAM: A Simple, Parameter-free Attention Module for Convolutional Neural Networks," in *Proc. 38th International Conference on Machine Learning*, 2021, pp. 11863–11874.
[31] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.
[32] S. Woo, J. Park, J. Lee *et al*., "CBAM: Convolutional Block Attention Module," in *Proc. European Conference on Computer Vision*, 2018, pp. 3–19.
[33] J. Gong, J. Xu, X. Tan *et al*., "Boundary-aware Geometric Encoding for Semantic Segmentation of Point Clouds," in *Proc. AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 1424–1432.
[34] W. Zhao, R. Zhang, Q. Wang *et al*., "BFANet: Revisiting 3D Semantic Segmentation with Boundary Feature Analysis," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025, pp. 29395–29405.
[35] S. Du, N. Ibrahimli, J. Stoter *et al*., "Push-the-boundary: Boundary-aware Feature Propagation for Semantic Segmentation," in *Proc. 2022 International Conference on 3D Vision*, 2022, pp. 124–133.
[36] S. Borse, Y. Wang, Y. Zhang *et al*., "IngerseForm: A Loss Function for Structured Boundary-aware Segmentation," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5901–5911.
[37] C. Wang, Y. Zhang, M. Cui *et al*., "Active Boundary Loss for Semantic Segmentation," in *Proc. AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, 2022, pp. 2397–2405.
[38] Z. Liu, P. Luo, X. Wang, *et al*., "Deep Learning Face Attributes in the Wild," in *Proc. IEEE International Conference on Computer Vision*, 2015, pp. 3730–3738.
[39] T. Karras, T. Aila, S. Laine *et al*., "Progressive Growing of GANs for Improved Quality, Stability, and Variation," arXiv Preprint, arXiv:1710.1019, 2018. https://arxiv.org/abs/1710.10196
[40] P. Wang, Y. He, X. Lv *et al*., "PartNeXt: A Next-generation Dataset for Fine-grained and Hierarchical 3D Part Understanding," arXiv Preprint, arXiv:2510.20155, 2025. https://arxiv.org/abs/2510.20155
[41] Pointcept Contributors. (2023). Pointcept: Point Cloud Perception Codebase. [Online]. Available: https://github.com/Pointcept/Pointcept
[42] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.
[43] J. Demsar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

Copyright © 2026 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).