APSIPA Transactions on Signal and Information Processing, 2025, 14, e104

Original Paper

JointFormer: Joint-Enhanced 3D Human Point Cloud Completion Based on Transformer

Min Zhou, Jieyu Chen, Xinpeng Huang and Ping An*

Key laboratory of Specialty Fiber Optics and Optical Access Networks, Joint International Research Laboratory of Specialty Fiber Optics and Advanced Communication, School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China

ABSTRACT

Human point cloud completion is a challenging yet indispensable task, devoted to filling missing parts in the collected incomplete point clouds. Existing methods overly rely on features extracted from surface points, neglecting the intrinsic joints information point clouds possess. To address this problem, we propose a new network with an encoder-decoder framework, named JointFormer. Firstly, we design a joint-enhanced encoder that provides more prior guidance on the overall structure of the partial input. Then, a generator is employed to generate sparse but complete point clouds. Finally, a decoder refines the rough point clouds into complete and dense human body point clouds in a coarse-to-fine manner. Moreover, combining transformer with the Convolutional Block Attention Module (CBAM), we design the Channel-Spatial Attention Transformer (CSAT) to better capture point cloud spatial relationships. Quantitative and qualitative evaluations demon-

*Corresponding author: Ping An, anping@shu.edu.cn. This work was supported in part by the National Natural Science Foundation of China under Grants of 62020106011, and 62071287, 62371278.

Received 22 May 2024; revised 05 August 2024; accepted 25 August 2024

ISSN 2048-7703; DOI 10.1561/116.20240041

strate that JointFormer outperforms the state-of-the-art completion method on our two human body point cloud datasets.

Keywords: Point cloud completion, human point cloud, transformer, joints estimation, spatial attention.

1 Introduction

Driven by the rapid development of 3D sensors, point clouds are emerging as an efficient data format for representing objects. Nonetheless, real-world point clouds often suffer from incompleteness and sparsity due to occlusion and sensor limitations. Therefore, point cloud completion has garnered widespread attention in computer vision and graphics, aiming to infer missing parts and densify sparse point clouds. This process benefits downstream tasks like human body reconstruction, object recognition, and pose estimation [11, 9, 2].

Compared to typical point cloud completion tasks, human point cloud completion poses more intricate challenges due to pose variability and complicated geometric relationships. The diverse poses of the human body demand highly adaptable algorithms to achieve coherent and realistic reconstruction across different postures. Moreover, the geometric relationships in human body point clouds, such as symmetry, are more nuanced than those in rigid objects, complicating the task of precise detail preservation and completion. In addition, the emergence of adversarial attack methods [3] for point cloud completion models has raised expectations for the robustness and resistance to interference of completion methods, which will enhance the security of downstream tasks such as 3D recognition and segmentation.

In recent years, deep learning-based point cloud completion has been flourishing. Some pioneering works on point cloud completion [20, 31, 4] directly handle 3D point cloud coordinates with an encoder-decoder architecture to generate complete point clouds. To avoid the information loss caused by max pooling operations in this architecture, some methods [29, 34, 30] incorporate Transformers to obtain detailed global features. However, when tackling non-rigid human point clouds with a body part such as a leg or arm severely missing, the global features extracted by existing methods are insufficient to infer the complete body parts.

Besides global features, another issue stems from the design of attention mechanisms for point clouds. Attention mechanisms have proven effective for point cloud analysis [33, 25]. Consequently, many recent methods have applied Transformers to learn structural features and long-range relationships within local regions of point clouds [34, 22, 30]. In these methods, the relationships between 3D points are usually explicitly introduced through position encoding.

However, spatial relationships introduced solely by position encoding seem insufficient. This affects the establishment of semantic relationships between points within local patches of the point cloud.

To address the aforementioned issues, we propose a novel human point cloud completion network called JointFormer. Our encoder is designed with a joint-enhanced strategy and consists of two branches: (i) encoding incomplete input to obtain local features and global features, (ii) encoding joints predicted from incomplete input to derive prior guidance for the complete point cloud. Secondly, we design a channel-spatial attention Transformer (CSAT) to incorporate positional information of features into the computation of point attention scores, extending the original channel attention and capturing more comprehensive local point cloud semantic information. Our main contributions can be summarized as follows:

1. We design a dual-branch joint-enhanced encoder. It predicts joints and further yields refined global features and local features, which introduces human pose prior structural information for the subsequent decoding phase.

2. We devise a channel-spatial attention Transformer (CSAT), which can better extract regional point cloud semantic relationships during both encoder and decoder in a simple yet effective way.

3. We evaluate our network on two self-crafted human point cloud datasets and public dataset PCN, which demonstrates our method achieved excellent performance.

2 Related Work

2.1 Point Cloud Completion and Reconstruction

Early works [13, 19, 18] convert irregular point clouds into 3D voxels and then perform point cloud completion based on 3D convolutions. However, this conversion requires significant computational resources and inevitably leads to geometric information loss. Consequently, in recent years, approaches that directly operate on point clouds have become mainstream. With the success of Transformer in Natural Language Processing (NLP), there has also been a surge of research exploring the potential of Transformer in point clouds. Therefore, we categorize related works into two groups: Non-Transformer-based methods and Transformer-based methods.

2.1.1 Non-Transformer-based Method

PointNet [16] and its successor, PointNet++ [17] have pioneered the application of deep learning directly on point clouds. In point cloud completion, PCN [31] firstly proposes using an encoder-decoder architecture, demonstrating that point-based completion methods have higher generalization performance and robustness compared to voxel-based methods. Pf-net [4] combines GAN with a hierarchical decoding strategy to predict missing parts. Moreover, Pmp-net [21] simulates the movement of points during completion and constrains the total distance of points to obtain a point-wise unique motion path. HyperCD [8] proposes measuring point cloud distances in hyperbolic space instead of Euclidean space, which can alleviate the vulnerability of CD to outliers. Flattening-Net [32] converts irregular 3D point clouds into regular 2D point geometry images, serving as a versatile representation for reconstruction and other tasks, and providing a novel perspective on point cloud completion.

2.1.2 Transformer-based Method

PoinTr [29] explicitly models the local geometric relationships of point clouds with Transformer, better learning and preserving structural information of point cloud. SnowflakeNet [26] leverages skip-transformer mechanism to infer the splitting patterns of the current layer from those of preceding layers within the SPD process. Seedformer [34] introduces patch seeds and extends Transformer to point generation operation, completing point cloud in a coarse-to-fine manner. Pmp-net++ [22] enhances the learning of point features with Transformer. Cross-PCC [24] explores the unsupervised method assisted by single-view images, and also leverages Transformer to capturing point relationships in 3D feature extraction, which provides valuable insights for point cloud completion without a large labeled dataset. ProxyFormer [6] introduces point proxy representation and design a missing part-sensitive Transformer, enabling the network to better predict missing parts. However, when applied to human point clouds with severe deficiencies, the performance of these methods seems mediocre due to the lack of internal structural relationship exploration.

2.2 Deep Learning on Human Point Cloud

In recent years, there has been a growing body of research on human point clouds. Some methods [35, 7, 23, 1, 11] propose modeling the complex surface structure directly from point clouds and then achieving an accurate action recognition and pose estimation of the human body. There are also methods dedicated to addressing the challenges of human reconstruction starting from point clouds. The previous method [5] extracts and maps skeleton features

with PointNet++ [17], then regresses to obtain SMPL [10] parameters for human shape reconstruction. VoteHMR [9] is proposed to recover human mesh from point clouds, which can effectively encode human body geometric information and has strong robustness to noisy inputs with self-occlusion and missing areas. It is evident that human point clouds play a crucial role in pose estimation and reconstruction. Human point clouds are clearly vital for pose estimation and reconstruction. Consequently, our research aims to address the issue of incomplete human point clouds to improve performance in these downstream tasks mentioned above.

3 Proposed Method

3.1 Overview

The overall architecture of JointFormer is illustrated in Figure 1, which consists of three parts: joint-enhanced encoder, generator, and decoder.

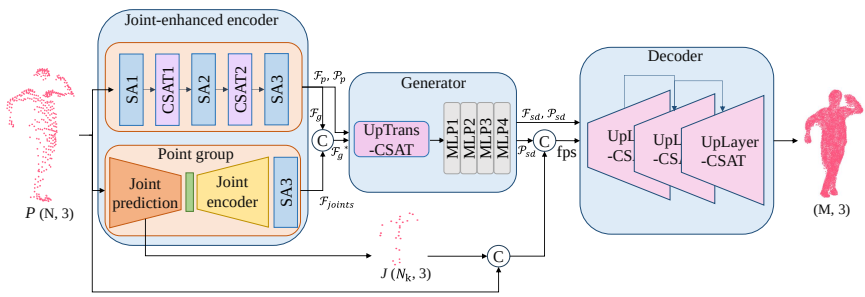


Figure 1: The overall architecture of JointFormer. The dual-branch Joint-enhanced encoder is applied to predict joints from partial input and derive fine-grained features. Then the generator produces sparse yet complete point clouds. Finally, the decoder gradually upsample to obtain dense and detailed output. Our devised CSAT is applied in those purple parts within the whole framework.

Our joint-enhanced encoder consists of a surface point encoding branch and a joint encoding branch. Given an incomplete input P, the former branch extracts global features $\mathcal{F}_g$, local features $\mathcal{F}_p$, and corresponding patch center $\mathcal{P}_p$. In the joint encoding branch, with generated point-wise votes $\{s_i, o_i, f_i\}$, where i represents the i-th point, aggregating those with the same semantic scores s_i, the human joints J are produced. Afterward, clustering partial input in terms of obtained joints, an ordered point patch sequence is obtained and fed into the Transformer block to extract joint features $\mathcal{F}_{joints}$. Concatenating $\mathcal{F}_{joints}$ with the global features $\mathcal{F}_g$ mentioned above, we obtain fine-grained global features $\mathcal{F}_g{}^* \in \mathbb{R}^{2\times N_g \times 1}$.

In generator, we feed $\mathcal{P}_p$, $\mathcal{F}_p$ and $\mathcal{F}_g^*$ into UpTrans-CSAT, while CSAT is applied to better capture the relationships among features. After a few MLPs, a rough yet complete seed point $\mathcal{P}_{sd}$ can be generated.

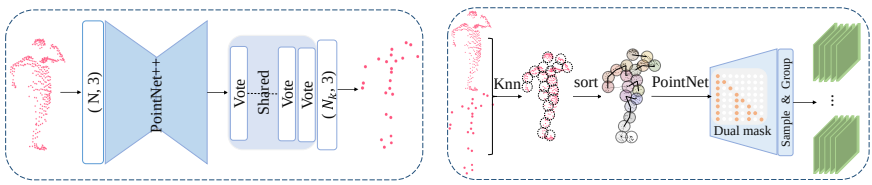
The decoder is devoted to recovering the complete point cloud reliably with fine details. We combine the upsampling process [34] and CSAT, named UpLayer-CSAT. Specifically, each layer interpolates seed points with seed features to obtain upsampled point cloud through CSAT. During all three layers, a finer point cloud $P_i(\ i = 1, 2, 3)$ is generated while the previous output serves as its input. Notably, to better preserve the detail in input and fully utilize predicted joint information, the entire decoder takes $P_0 \in \mathbb{R}^{N_0 \times 3}$ as input, which is obtained by concatenating P, J, and $\mathcal{P}_{sd}$.

3.2 Joint-enhanced Encoder

3.2.1 Joint Prediction

For input incomplete human point cloud $P \in \mathbb{R}^{N \times 3}$, the joint prediction module generates human joints $J \in \mathbb{R}^{N_k \times 3}$, adhering to SMPL (Bogo et al., 2016), where N_k is 24.

Specifically, as shown in Figure 2(a), we employ PointNet++ [17] as the backbone to extract joint feature f_{joint}.



(a) Joint prediction (b) Joint encoder

Figure 2: Modules in the Joint-enhanced Encoder: Detailed illustrations of the prediction of joints (a) with votes and the architecture of the joint encoder (b) for handling human joints around the partial input.

We leverage voting module [15, 9] to generate point-wise votes $\{s_i, o_i, f_i\}$ due to its success on predicting offset coordinates and features for each seed point, which will benefit the joints discovery even with severe noises, missed areas, and occlusions.

The module consists of MLPs, followed by independent heads for body part segmentation, joint regression, and feature updating respectively. s_i is formed by the action of an FC layer and softmax. The joint regression head adopts an FC layer to output offsets $o_i \in \mathbb{R}^3$ for each point p_i, and also aggregates more reasonable contextual information to define joint positions.

The feature updating head uses residual connections to update vote features $f_i = f_{joint_i} + \Delta f_i, i = 1,,N$. Δf_i is extracted from a shared MLP with an FC layer.

Once we have obtained the point-wise votes, we can proceed to the group and aggregate them to obtain joints. Specifically, the coordinates j_k of each joint can be represented as:

$$j_k = \frac{1}{\sum_{i=1}^{N}(s_i)^k} \sum_{i=1}^{N} (p_i + o_i)(s_i)^k, \quad k = 1,,24 \tag{1}$$

Where s_i represents semantic content indicating the body part, p_i represents each point of the partial input.

Throughout joint estimation, the point-wise labels y_i can provide auxiliary support. Practically, we set all labels to a specific value. This ensures that the generated points align with the input human point cloud in the same coordinate system while also promoting model convergence. Moreover, the predicted joint coordinates are regularized against the ground truth with L2 loss during training.

3.2.2 Joint Encoder

As shown in Figure 2(b), given input incomplete point cloud $P \in \mathbb{R}^{N\times 3}$ and predicted joint $J \in \mathbb{R}^{N_k\times 3}$, firstly, we partition input incomplete point cloud with KNN to obtain the neighbors of each joint $J_{neigh} \in \mathbb{R}^{N_k\times K\times 3}$. Then, we compute the distances among joints, followed by sorting to determine the order of joints. Finally, iterating through all 24 joints, the index tensor J_{idx} labeled with joint order information is generated, while also sorting $J_{neigh} \in \mathbb{R}^{N_k\times K\times 3}$ accordingly.

Having obtained the ordered joint coordinates J and the ordered joint neighbors J_{neigh}, we first encode J_{neigh} using PointNet [16] to derive the features of joint neighbors. Then, we perform attention calculation on this ordered sequence of point features with a dual masking strategy, which masks some previous tokens of the current token. It can be represented as follows:

$$\text{SelfAttention}(\boldsymbol{T}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{D}} - \left(1 - \boldsymbol{M^d}\right)\cdot\infty\right)\boldsymbol{V} \tag{2}$$

where Q, K, and V are obtained by encoding T with different weights along the channel dimension D. The masked positions M^d are set to 0 if masked, and 1 otherwise.

Eventually, our joint encoder is composed entirely of transformer decoder blocks with a dual masking strategy, obtaining joint feature $\mathcal{F}_{\text{joints}}$.

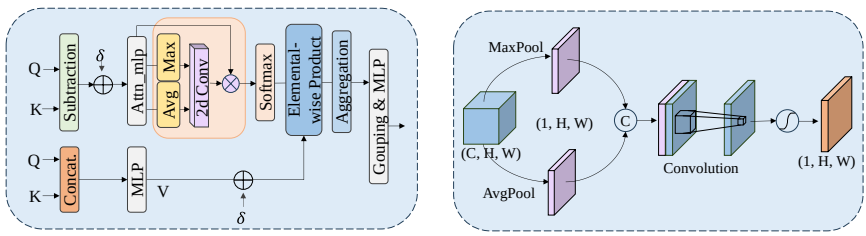
3.3 Channel-Spatial Attention Transformer

CSAT complements the point attention mechanism by excavating relationships within attention weights in both channel and spatial dimensions.

Given seed points $\mathcal{P}_{sd} \in \mathbb{R}^{N_{sd} \times 3}$ and corresponding seed features $\mathcal{F}_{sd} \in \mathbb{R}^{N_{sd} \times D}$, the query vector Q are generated by concatenating them and passing through a MLP. The key vector K in UpLayer-CSAT is composed of output features from the previous layer to retain features from input. Then, the value vector V are composed of concatenated Q and K, as shown in Figure 3(a). CSAT calculates channel-wise attention weights $\hat{a}_{ij}$ after subtracting each point from its k nearest neighbors $\mathcal{N}(i)$ as follows:

$$\hat{a}_{ij} = \alpha(\beta\left(Q\right) - \gamma\left(K\right) + \delta),\ j \in \mathcal{N}(i) \tag{3}$$

where α, β, and γ are feature mapping functions MLP and linear layers. $Q, K \in \mathbb{R}^{D_h \times N_l \times K}$ represents fused features of joint-wise and point-wise features.



(a) CSAT (b) Spatial attention

Figure 3: The architecture of CSAT and spatial attention. The orange part in (a)CSAT denotes our spatial attention (b) insert in point vector attention mechanism.

Then, as shown in Figure 3(b), two pooling operations is utilized to generate average-pooled features $\mathcal{F}_{\text{avg}} \in \mathbb{R}^{1 \times H \times W}$ and max-pooled features $\mathcal{F}_{\max} \in \mathbb{R}^{1 \times H \times W}$ across the channel in $\hat{a}_{ij}$, which are then concatenated and fed into a convolution layer. We compute point attention weights $\hat{a}^*_{ij}$ as follows:

$$\hat{a}^*_{ij} = \sigma\left(f^{7\times7} Cat\left(\text{Avg}\left(\hat{a}_{ij}\right), \text{Max}\left(\hat{a}_{ij}\right)\right) * \hat{a}_{ij}\right) \tag{4}$$

where σ denote sigmoid function and $f^{7\times7}$ represents a convolution operation with the filter size of 7×7.

Through elemental-wise multiplication of computed attention weights α_{ij} and value vector, we obtain the features h_i of new points around each seed point.

$$h_i = \sum_{j \in \mathcal{N}(i)} \exp\left(\hat{a}_{ij}\right) * \left(\psi\left(V\right) + \delta\right) \tag{5}$$

where ψ is a feature mapping function. exp() denotes softmax function used to normalize the computed weights $\hat{a}_{ij}$. $V \in \mathbb{R}^{D_h \times N_l \times K}$ represents point-wise features.

The aggregated features of the K-nearest neighbors of all seed points generate the upsampled features, doubling as the key vectors for the subsequent iteration in the skip connection. Finally, a set of point offsets is obtained using a shared multi-layer perceptron based on the generated new point features. Adding these offsets to the replicated original points results in the generation of new upsampling points.

3.4 Loss Function

We propose a combined loss of Chamfer Distance (CD) and joint L2 loss in our end-to-end human point cloud completion network, defined as:

$$\mathcal{L} = \lambda_{cd}\mathcal{L}_{CD} + \lambda_{joint}\mathcal{L}_{Joint} \tag{6}$$

where λ_{cd}, λ_{joint} represent the weights of hierarchical CD loss and joint estimation loss respectively. In this paper, we set λ_{cd} to 1 and λ_{joint} to 0.5.

Specifically, generated point clouds at multiple stages of point cloud completion process are supervised, denoted as $\mathbb{P} \in \{\mathcal{P}_{seed}, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3\}$, referred to as hierarchical CD loss $\mathcal{L}_{CD}$. Ground truth $\mathcal{P}_{gt} \in \mathbb{R}^{N_t \times 3}$ is aligned with the generated point cloud in resolution with Farthest Point Sampling (FPS). The $\mathcal{L}_{CD}$ can be expressed as:

$$\mathcal{L}_{CD} = \sum_{\mathcal{P}_i \in \mathbb{P}} CD(\mathcal{P}_i, \mathrm{FPS}(\mathcal{P}_{gt})) \tag{7}$$

Additionally, to better utilize joint features obtained based on joint estimation, we incorporate joint loss into the network to supervise the output of the joint estimation module. The joint estimation loss $\mathcal{L}_{Joint}$ is formulated as:

$$\mathcal{L}_{Joint} = \frac{1}{N}\|pred - gt\|_2^2 \tag{8}$$

4 Experiments

4.1 Evaluation Metrics and Implementation Details

We utilize widely adopted metrics including Chamfer Distance (CD), Chamfer Distance with L1 norm (CD-ℓ1), and F-score.

CD evaluates the overall distribution, which ensures generated point cloud represents the surface of the target. To alleviate the sensitivity of the L2 norm to individual outliers, CD-$\ell 1$ is adopted as a more balanced metric. Additionally, the F-Score assesses the similarity between the generated and the the target.

In the joint-enhanced encoder module, we set the point-wise labels y_i to be 0.001, the value of k in KNN to be 32, and the groups of KNN N_k to be 24. We conducted extensive experiments on several NVIDIA RTX4090 GPUs. We trained our JointFormer end-to-end on PyTorch for 400 epochs. We used the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

4.2 Datasets for Human Point Cloud Completion

Given the absence of a publicly available dataset of partial human body point clouds, we referred to [14] to create two high-quality multi-view incomplete human body point cloud datasets: THuman2.0 [28] and AMASS [12].

THuman2.0: To simulate the scenarios of incompleteness and sparsity that occur in the real world, we first rendered 10 depth images from different random viewpoints based on the model. Subsequently, transforming the generated depth images into the world coordinate system, we obtain partial point cloud data. Figure 4 shows an example of point cloud data obtained with a focal length of 100.

Finally, the THuman2.0 dataset consists of 4200 training data, 530 validation data, and 530 test data. Each ground truth point cloud is generated by evenly sampling 16384 points from the model surface. During experiments, the batch size is 8, and the initial learning rate is 0.001 and decays by 0.1 every 100 epochs.

AMASS-part: Considering the limited variety of human models and poses in the THuman2.0 dataset, we further developed a new, more diverse dataset from part of AMASS [12], named AMASS-part. AMASS encompasses a rich collection of human motion sequences stored as SMPL parameters. Therefore, we first convert the SMPL parameters into 3D human models and then generate point cloud data using the same approach as in THuman2.0 mentioned above.

Finally, AMASS-part consists of 4489 human models. After sampling from 10 random viewpoints, the training set contains 43,830 samples, while the test set and the validation set each have 5,530 samples. We raised the batch size to 48; the initial learning rate is 0.0005.

4.3 Evaluation on THuman2.0 dataset

We compared our JointFormer with several baseline methods on our THuman2.0 dataset. For fairness, we retrained these methods with their optimal parameters respectively.

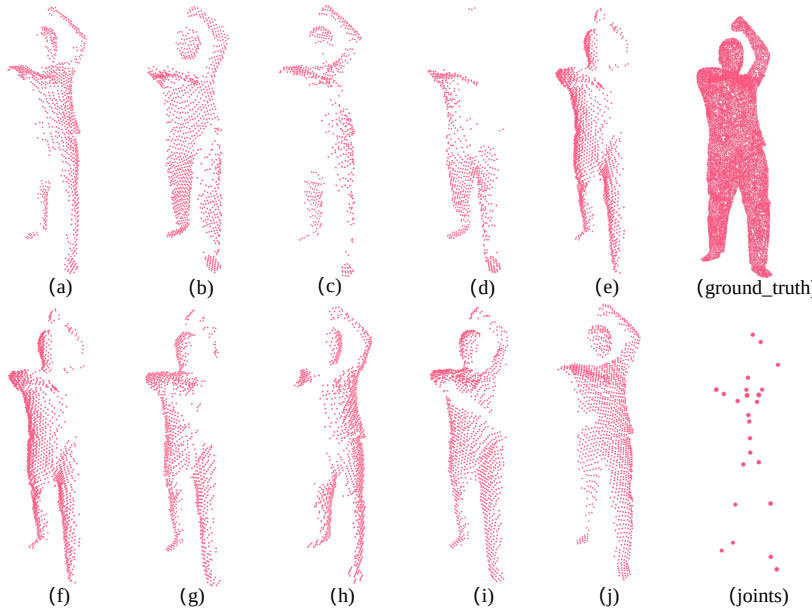


Figure 4: Example of our THUman2.0 dataset. (a)-(j) respectively represent partial human point clouds sampled from 10 random viewpoints. The resolution of ground_truth is 16384. The joints are extracted from SMPL parameters.

4.3.1 Quantitative Results

As shown in Table 1, we report Chamfer Distance, Chamfer Distance with L1 norm, and F-Score with several previous methods. Generally, lower Chamfer Distance indicates more accurate reconstructive shape. Compared to the second-ranked AdaPoinTr, JointFormer reduces CD-ℓ1 by 0.48, which is 8.6% lower, while CD is 10.0% lower and F-Score is 4.5% higher. As indicated by results, the global guidance introduced through joint information in human poses and the exploration of spatial relationships indeed assist in the completion of incomplete human point clouds. Our JointFormer, through its novel designs, surpasses them by a significant margin.

4.3.2 Qualitative results

As shown in Figure 5, we can visually compare with several previous methods. The 1st column displays incomplete point clouds with varying degrees of missing data. The 1st point cloud lost data on the back and left leg due to

Table 1: Results on THuman2.0 in terms of Chamfer Distance Œ1000(lower is better), L1 Chamfer Distance Œ1000 (lower is better) and F-Score@1% (higher is better).

Methods	CD ↓	CD-ℓ1 ↓	F1 ↑
FoldingNet [27]	4.00	33.68	0.16
PCN [31]	0.37	9.55	0.66
PoinTr [29]	0.31	9.23	0.66
SnowflakeNet [26]	0.17	6.77	0.83
PMPNet++ [22]	0.30	8.74	0.70
Seedformer [34]	0.12	5.91	0.88
AdaPoinTr [30]	0.10	5.57	0.89
Ours	**0.09**	**5.09**	**0.93**

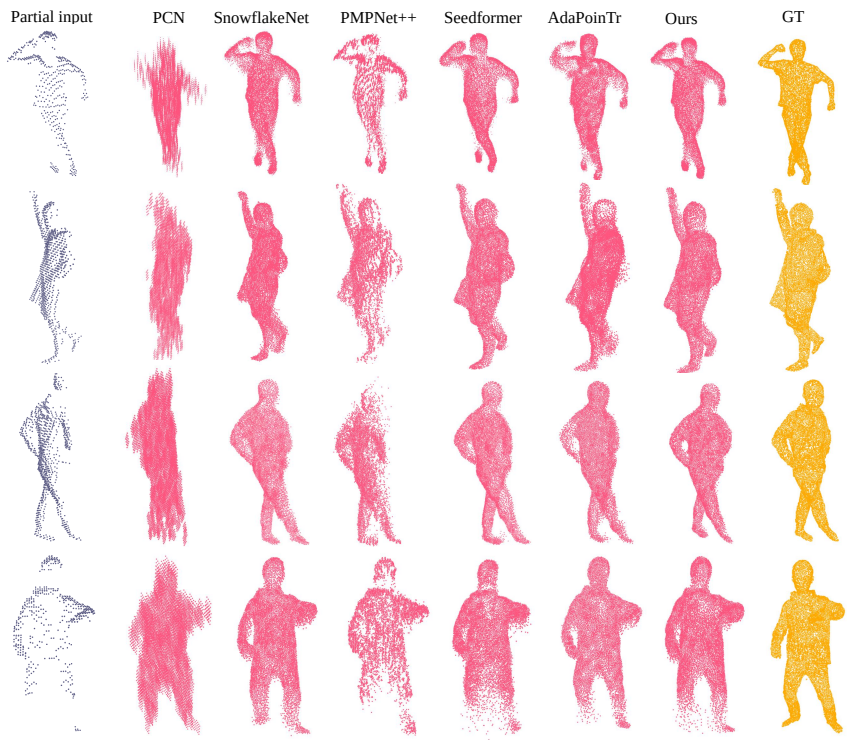


Figure 5: Four visual examples of completion results on the THuman2.0 dataset using various methods. Each row shows results generated by feeding the model with data from different camera angles: the 1st and 4th rows represent data captured from above the human, the 2nd row represents data from the front, and the 3rd row represents data from behind the human.

self-occlusion. The 2nd and 3rd point clouds miss back and front respectively. The 4th point cloud suffers significant data loss in the body and legs. Each row shows the results of different methods for completing the point clouds for each partial data. In the 1st and 2nd rows, our method has finer details on the right hand and neck of the human body, with a more even distribution of surface and fewer outliers. Especially, it can be observed that our JointFormer has the smoothest surface and provides a more detailed and credible completion of the left leg in the 1st row. In the 3rd row, facing the severely missing on the neck of the human body, our method has a more reasonable edge, while Seedformer and AdaPoinTr introduce some outliers at the toes and uneven distribution of the right arm. In the 4th row, where the legs are almost entirely missing, JointFormer performs better at detecting the presence of the right foot compared to other methods. Meanwhile, SnowflakeNet and AdaPoinTr, despite having fewer outliers, prematurely halt the completion process. In conclusion, our JointFormer has a stronger ability to restore details of human body parts and smooth surface points.

4.3.3 Comparison Under Different Levels of Difficulty

As shown in Table 2, we conduct experiments under different difficulty levels to further evaluate the performance of JointFormer. CD1-S, CD1-M, and CD1-H represent CD-ℓ1 when the viewpoint focal length is set to 100, 150, and 200. Considering that training data was generated at a focal length of 100, as the focal length increases, the discrepancy between the input and training data grows. Consequently, the difficulty of model inference increases. However, compared to previous methods, our model performs well at all levels and averages, except for slightly higher than PoinTr in hard level (focal length is 200). Compared to the second-ranked Seedformer, JointFormer decreased by 8.2% on CD-ℓ1 Avg. This indicates that our method exhibits superior generalization performance when applied to data not encountered during training.

Table 2: Results on THuman2.0 under *Simple, Medium, Hard* in terms of L1 Chamfer Distance Œ1000 (lower is better).

Methods	CD1-H↓	CD1-M↓	CD1-S↓	CD1-Avg↓
FoldingNet [27]	33.10	33.21	33.68	33.33
PCN [31]	15.39	11.55	9.55	12.16
PoinTr [29]	**8.96**	8.79	8.99	8.91
PMPNet++ [22]	14.11	9.41	8.70	10.74
Seedformer [34]	10.12	6.92	5.91	7.65
Ours	9.89	**6.09**	**5.09**	**7.02**

4.4 Evaluation on AMASS-part dataset

On the AMASS-part dataset, we also conduct experiments for our method and other five state-of-the-art methods. Compared to THuman2.0, the AMASS-part is sampled from human motion sequences and includes more complicated human postures, making the completion task significantly more challenging.

The results are shown in Table 3. We see our method also achieves the best performance in this more challenging dataset. Compared with Seedformer, not only the CD-$\ell 1$ performance drop of our method is up to 2.8, but also the CD and F1 performance much better. We also visualize the results in Figure 6 to show the excellent performance of our method on human motion sequences. Compared with Seedformer, in the 1-st row, our model delivers a more complete point cloud of the human head. In the 2-nd row, it provides a more accurate and reliable point cloud of the hands. In the 3-rd row, it produces a finer and more detailed point cloud of the left side of the body.

Table 3: Results on AMASS-part in terms of Chamfer Distance Œ1000 (lower is better), L1 Chamfer Distance Œ1000(lower is better)and F-Score@1% (higher is better).

Methods	CD ↓	CD-$\ell 1$ ↓	F1 ↑
FoldingNet [27]	3.26	31.16	0.11
PCN [31]	1.92	23.13	0.21
PoinTr [29]	2.46	25.12	0.28
PMPNet++ [22]	1.69	18.69	0.31
Seedformer [34]	1.79	16.92	0.49
Ours	**1.19**	**14.10**	**0.56**

4.5 Ablation Study

4.5.1 Analysis of joint-enhanced encoder

The evaluation results of the joint-enhanced encoder are shown in Table 4. The baseline model A is the basic point transformer for point cloud completion, which uses encoder-decoder architecture with point Transformer. In this model, we extract local and global features with SA layers, while the decoder consists of point transformer without spatial attention. We then add the joint encoding branch in the encoder (model B). We see the joint encoding branch improves the baseline by 0.13 in Chamfer Distance with L1-norm. Furthermore, we incorporate the joint regularization loss (model C) to investigate its impact on the completion results. It can be seen the addition of regularization loss slightly improves the completion performance. That is, the human joints predicted by a joint encoding branch can indeed provide reliable global information for guiding completion tasks, particularly when the input point cloud has substantial gaps or severe missing regions.

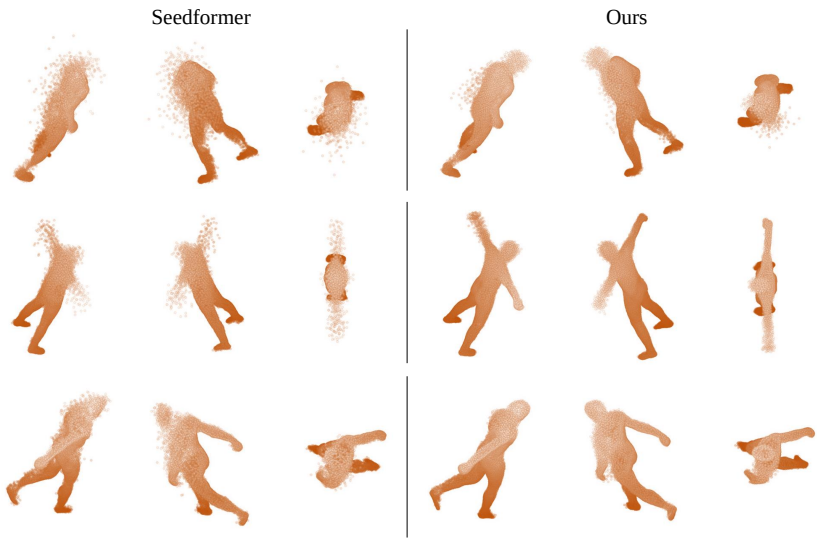


Figure 6: Three visual examples of human point cloud completion results on the AMASS-part dataset. The first two images in each group are side views, and the last one is the top view, which can provide a more intuitive evaluation of the completion results of the model.

Table 4: Ablation studies on the THuman2.0 dataset. We investigate the impact of CSAT and joint-enhanced encoder (joint-enc.) design on network performance.

Model	joint-enc.	joint-loss	CSAT	CD-ℓ1↓	F-Score@1% ↑
Seedformer				5.91	0.88
A				5.29	0.92
B	✓			5.16	0.93
C	✓	✓		5.15	0.93
D			✓	5.14	0.93
E	✓	✓	✓	5.09	0.93

4.5.2 Analysis of CSAT

To verify the effectiveness of our CSAT (Channel-Spatial Attention Transformer), we conducted ablation experiments as shown in Table 4. Model E represents our JointFormer. For comparison, we create model D by removing the joint encoding branch from the encoder. Notably, even without the joint encoding branch, model D still integrates CSAT into the surface point encoding branch, as well as into the upsampling Transformer blocks within

the generator and decoder. This leads to a decrease in the CD-ℓ1 score to 5.14 compared to model A (lower is better). The results confirm the effectiveness of our joint-enhanced encoder.

Additionally, we evaluated CSAT on the public dataset PCN. As shown in Table 5, our model performs better than previous methods in all 8 categories except for lamp. This indicates that the introduction of additional spatial attention does not bring good improvement effects for objects with the shape of a lamp. However, on average, the CSAT can better assist Transformers in capturing neighboring relationships in unordered and structurally irregular point clouds, which is crucial for point cloud completion tasks. Comparing Seedformer and Mode D, we can be seen that CSAT only introduces a slight improvement (0.03) on the PCN dataset, while there is a 13% improvement on the human point cloud dataset. This further validates the effectiveness of our CSAT.

Table 5: Analysis of Channel-Spatial Attention Transformer on the PCN dataset in terms of L1 Chamfer Distance Œ1000 (lower is better).

Methods	Average	Plane	Carbinet	Car	Chair	Lamp	Couch	Table	Boat
FoldingNet	14.31	9.49	15.80	12.61	15.55	16.41	15.97	13.65	14.99
PCN	9.64	5.50	22.70	10.63	8.70	11.00	11.34	11.68	8.59
GRNet	8.83	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04
PMP-Net	8.73	5.65	11.24	9.64	9.51	6.95	10.83	8.72	7.25
PoinTr	8.38	4.75	10.47	8.68	9.39	7.75	10.93	7.78	7.29
SnowflakeNet	7.21	4.29	9.16	8.08	7.89	6.07	9.23	6.55	6.40
Seedformer	6.74	3.85	9.05	8.06	7.06	**5.21**	8.85	6.05	5.85
Ours	**6.71**	**3.83**	**9.04**	**8.02**	**7.01**	5.26	**8.76**	**6.00**	**5.82**

5 Conclusion

In this paper, we propose JointFormer, a novel method for point cloud completion in non-rigid objects, particularly focusing on the human body. By fully leveraging self-attention mechanisms, our method effectively captures both local and long-range structural relationships among unordered points. In cases of severe missing situations, the joint encoding branch plays a crucial role in guiding the inference process. Extensive comparisons and ablation studies underscore the superiority of JointFormer, demonstrating its capability to outperform state-of-the-art methods. Additionally, we have introduced two new, and more challenging datasets specifically designed for human point cloud completion. Further exploration of our architecture in other 3D human reconstruction tasks could present an exciting direction for future research.

References

[1] Z. Cai, L. Pan, C. Wei, W. Yin, F. Hong, M. Zhang, C. C. Loy, L. Yang, and Z. Liu, "Pointhps: Cascaded 3d human pose and shape estimation from point clouds", *arXiv preprint arXiv:2308.14492*, 2023.

[2] S. Christen, W. Yang, C. Pérez-DArpino, O. Hilliges, D. Fox, and Y.-W. Chao, "Learning human-to-robot handovers from point clouds", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 9654–64.

[3] S. Hu, J. Zhang, W. Liu, J. Hou, M. Li, L. Y. Zhang, H. Jin, and L. Sun, "Pointca: Evaluating the robustness of 3d point cloud completion models against adversarial examples", in *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37, No. 1, 2023, 872–80.

[4] Z. Huang, Y. Yu, J. Xu, F. Ni, and X. Le, "Pf-net: Point fractal network for 3d point cloud completion", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, 7662–70.

[5] H. Jiang, J. Cai, and J. Zheng, "Skeleton-aware 3d human shape reconstruction from point clouds", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 5431–41.

[6] S. Li, P. Gao, X. Tan, and M. Wei, "Proxyformer: Proxy alignment assisted point cloud completion with missing part sensitive transformer", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, 9466–75.

[7] X. Li, Q. Huang, Y. Zhang, T. Yang, and Z. Wang, "PointMapNet: Point Cloud Feature Map Network for 3D Human Action Recognition", *Symmetry*, 15(2), 2023, 363.

[8] F. Lin, Y. Yue, S. Hou, X. Yu, Y. Xu, K. D. Yamada, and Z. Zhang, "Hyperbolic chamfer distance for point cloud completion", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, 14595–606.

[9] G. Liu, Y. Rong, and L. Sheng, "Votehmr: Occlusion-aware voting network for robust 3d human mesh recovery from partial point clouds", in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, 955–64.

[10] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: a skinned multi-person linear model", *ACM Transactions on Graphics (TOG)*, 34(6), 2015, 1–16.

[11] W. Ma, M. Yin, G. Li, F. Yang, and K. Chang, "PCMG: 3D point cloud human motion generation based on self-attention and transformer", *The Visual Computer*, 40(5), 2024, 3765–80.

[12] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "AMASS: Archive of motion capture as surface shapes", in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, 5442–51.

[13] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition", in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 2015, 922–8.

[14] L. Pan, X. Chen, Z. Cai, J. Zhang, H. Zhao, S. Yi, and Z. Liu, "Variational relational point completion network", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, 8524–33.

[15] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3d object detection in point clouds", in *proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 9277–86.

[16] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, 652–60.

[17] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space", *Advances in neural information processing systems*, 30, 2017.

[18] G. Riegler, A. Osman Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, 3577–86.

[19] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, 1746–54.

[20] L. P. Tchapmi, V. Kosaraju, H. Rezatofighi, I. Reid, and S. Savarese, "Topnet: Structural point cloud decoder", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, 383–92.

[21] X. Wen, P. Xiang, Z. Han, Y.-P. Cao, P. Wan, W. Zheng, and Y.-S. Liu, "Pmp-net: Point cloud completion by learning multi-step point moving paths", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, 7443–52.

[22] X. Wen, P. Xiang, Z. Han, Y.-P. Cao, P. Wan, W. Zheng, and Y.-S. Liu, "Pmp-net++: Point cloud completion by transformer-enhanced multi-step point moving paths", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1), 2022, 852–67.

[23] Z. Weng, A. S. Gorban, J. Ji, M. Najibi, Y. Zhou, and D. Anguelov, "3d human keypoints estimation from point clouds in the wild without human labels", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 1158–67.

[24] L. Wu, Q. Zhang, J. Hou, and Y. Xu, "Leveraging single-view images for unsupervised 3D point cloud completion", *IEEE Transactions on Multimedia*, 2023.

[25] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer v2: Grouped vector attention and partition-based pooling", *Advances in Neural Information Processing Systems*, 35, 2022, 33330–42.

[26] P. Xiang, X. Wen, Y.-S. Liu, Y.-P. Cao, P. Wan, W. Zheng, and Z. Han, "Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer", in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, 5499–509.

[27] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, 206–15.

[28] T. Yu, Z. Zheng, K. Guo, P. Liu, Q. Dai, and Y. Liu, "Function4d: Real-time human volumetric capture from very sparse consumer rgbd sensors", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, 5746–56.

[29] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "Pointr: Diverse point cloud completion with geometry-aware transformers", in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, 12498–507.

[30] X. Yu, Y. Rao, Z. Wang, J. Lu, and J. Zhou, "AdaPoinTr: Diverse Point Cloud Completion With Adaptive Geometry-Aware Transformers", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12), 2023, 14114–30.

[31] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "Pcn: Point completion network", in *2018 international conference on 3D vision (3DV)*, 2018, 728–37.

[32] Q. Zhang, J. Hou, Y. Qian, Y. Zeng, J. Zhang, and Y. He, "Flattening-net: Deep regular 2d representation for 3d point cloud analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8), 2023, 9726–42.

[33] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer", in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, 16259–68.

[34] H. Zhou, Y. Cao, W. Chu, J. Zhu, T. Lu, Y. Tai, and C. Wang, "Seedformer: Patch seeds based point cloud completion with upsample transformer", in *European conference on computer vision*, 2022, 416–32.

[35] Y. Zhou, H. Dong, and A. El Saddik, "Learning to estimate 3d human pose from point cloud", *IEEE Sensors Journal*, 20(20), 2020, 12334–42.