3D Computer Vision and Visual Data Analysis

Doctor of Philosophy (Computer Science)

THE UNIVERSITY OF
SYDNEY

Supervisor: A/Prof. Weidong Cai
Associate Supervisor: Dr. Yang Song

A thesis submitted in fulfilment of
the requirements for the degree of
Doctor of Philosophy

School of Computer Science
Faculty of Engineering
The University of Sydney
Australia

11 January 2024

Declaration

This is to certify that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Jianhui Yu
10-Jan-2024

Abstract

This thesis delves into the domain of 3D data analysis, an area of immense significance in fields including computer graphics, virtual reality, and medical imaging. While 2D data has been extensively studied in computer vision, 3D data introduces an additional layer of complexity, either due to an added spatial dimension or a temporal aspect in video data. This research focuses on three forms of 3D data: point clouds, human meshes, and face videos.

In point cloud analysis, we focus on key tasks including classification, segmentation, and semantic segmentation. We first investigate medical point clouds, where we propose a transformer-based model with a novel attention mechanism and a graph reasoning module for classification and segmentation tasks. We also introduce a method for rotation-invariant feature learning, improving analysis robustness and computational efficiency.

Moving to 3D human modeling, our work explores text-guided human texture generation. Traditional 3D modeling techniques often fall short in capturing the nuanced textural details of human models. We use a deep learning framework, combining diffusion generative models with physically based rendering and a 3D coordinate network. This method generates high-quality textures and ensures they align semantically with input texts.

In the realm of face video data, we begin by proposing a generative adversarial network pipeline for synthesizing faces and predicting micro-expression labels. We also introduce a large-scale face video dataset, complete with textual descriptions, and present a novel text-to-face generation model using bidirectional transformers and an innovative video token technique. Our experiments demonstrate both the superiority of our method and the high-quality face dataset.

Overall, this thesis contributes significantly to 3D data processing, showing great potential in point cloud analysis, 3D human modeling, and face video processing, promising research and practical advancements.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, A/Prof. Weidong Cai, for his invaluable guidance, support, and constructive criticism throughout the course of my project. During my PhD study, I have came across many difficulties, and my supervisor always shares his support and care about me and my family. I would like to thank my supervisor for his immense knowledge and expertise in the field, combined with the constant encouragement to push my boundaries and explore new research fields and novel ideas. All of these have contributed greatly to my professional growth and to the successful completion of this work. I am truly thankful for the opportunity to learn from such an inspiring mentor. I am also grateful to my associate supervisor, Dr. Yang Song, for all of her professional and insightful suggestions and help on my research works.

I would like to show my gratitude to all the colleagues that I work with: Chaoyi Zhang, Dr. Dongnan Liu, Zihao Tang, Heng Wang, and Dr. Yuqian Chen. I greatly appreciate the care and help, as well as the guidance that they provide during my PhD research study.

I want to express my heartfelt thanks to my beloved family for their love, patience, and belief in my abilities. My parents have provided me with unwavering support and endless motivation, standing by me in the most challenging of times. Their faith in my potential and their invaluable advice have been my pillars of strength.

Lastly, I would like to thank my girlfriend, Shiwen Lao, for being my rock, for her understanding and patience during the countless late nights and stressful days. Your love, patience, and unwavering belief in me have made all the difference.

To all of you, I owe more than words can convey. This achievement would not have been possible without your enduring love, support, and encouragement. Thank you.

Publications

[1] **Yu, J.**, Zhu, H., Jiang, L., Loy, C. C., Cai, W., and Wu, W. (2024). PaintHuman: Towards High-fidelity Text-to-3D Human Texturing via Denoised Score Distillation. Accepted by *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI).*

[2] **Yu, J.**, Zhu, H., Jiang, L., Loy, C.C., Cai, W. and Wu, W. (2023). CelebV-Text: A Large-Scale Facial Text-Video Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 14805-14814).

[3] **Yu, J.**, Zhang, C. and Cai, W. (2023). Rethinking Rotation Invariance with Point Cloud Registration. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)* (pp. 3313-3321).

[4] Zhang, D.*, **Yu, J.***, Zhang, C., and Cai, W. (2023). PaRot: Patch-Wise Rotation-Invariant Network via Feature Disentanglement and Pose Restoration. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)* (pp. 3418-3426).

[5] Wang, H., Zhang, C., **Yu, J.**, and Cai, W. (2022). Spatiality-guided Transformer for 3D Dense Captioning on Point Clouds. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 1393-1400).

[6] **Yu, J.**, Zhang, C., Song, Y. and Cai, W. (2021). ICE-GAN: identity-aware and capsule-enhanced GAN with graph-based reasoning for micro-expression recognition and synthesis. In *2021 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8).

[7] **Yu, J.**, Zhang, C., Wang, H., Zhang, D., Song, Y., Xiang, T., Liu, D. and Cai, W. (2021). 3d medical point transformer: Introducing convolution to attention networks for medical point cloud analysis. arXiv preprint arXiv:2112.04863.

[8] Xiang, T., Zhang, C., Song, Y., **Yu, J.** and Cai, W. (2021). Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 915-924).

[9] Zhang, C., **Yu, J.**, Song, Y. and Cai, W. (2021). Exploiting edge-oriented reasoning for 3d point-based scene graph analysis. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition* (pp. 9705-9715).

[10] Wang, H., Zhang, C., **Yu, J.**, Song, Y., Liu, S., Chrzanowski, W. and Cai, W. (2021). Voxel-wise cross-volume representation learning for 3d neuron reconstruction. In *Machine Learning in Medical Imaging: 12th International Workshop (MLMI)*, (pp. 248-257).

[11] Wang, H., Song, Y., Zhang, C., **Yu, J.**, Liu, S., Peng, H. and Cai, W. (2021). Single neuron segmentation using graph-based global reasoning with auxiliary skeleton loss from 3D optical microscope images. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)* (pp. 934-938).

Authorship Attribution Statement

Chapter 2 of this thesis is published as [7].
I am the first author of the publication [7]. I designed this study, analysed the data and wrote the drafts with the co-authors.

Chapter 3 of this thesis is published as [3].
I am the first author of the publication [3]. I designed this study, analysed the data and wrote the drafts with the co-authors.

Chapter 4 of this thesis is published as [1].
I am the first author of the publication [1]. I designed this study, analysed the data and wrote the drafts with the co-authors.

Chapter 5 of this thesis is published as [6].
I am the first author of the publications [6]. I designed this study, analysed the data and wrote the drafts with the co-authors.

Chapter 6 of this thesis is published as [2].
I am the first author of the publications [2]. I designed this study, analysed the data and wrote the drafts with the co-authors.

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Jianhui Yu
10-Jan-2024

As supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Weidong Cai
10-Jan-2024

Contents

Declaration ii

Abstract iii

Acknowledgements iv

Publications v

Authorship Attribution Statement vii

Contents ix

List of Figures xiv

List of Tables xix

Chapter 1 Introduction 1

1.1 3D Point Clouds 2

1.1.1 Challenges 2

1.1.2 Medical Point Clouds 3

1.1.3 Rotated Point Clouds 4

1.2 3D Human Meshes 5

1.2.1 Challenges 6

1.2.2 3D Human Texture Generation 6

1.3 Face Videos 9

1.3.1 Challenges 9

1.3.2 Micro-Expression Face Video 10

1.3.3 Text-guided Face Video Generation 11

1.4 Thesis Contributions and Organization 12

Chapter 2 Deep Learning-based Analysis for Medical Point Clouds 15

2.1 Introduction . . . 15

2.2 Literature Review . . . 16

2.2.1 3D General Point Clouds . . . 16

2.2.2 3D Medical Point Clouds . . . 18

2.2.3 Transformer . . . 19

2.2.4 Graph-based Reasoning . . . 21

2.3 Methods . . . 22

2.3.1 Local Context Augmentation . . . 23

2.3.2 Convolution at Query . . . 23

2.3.3 Relative Positional Embedding . . . 25

2.3.4 Multi-Graph Reasoning . . . 26

2.4 Experiments and Results . . . 26

2.4.1 Datasets . . . 27

2.4.2 Evaluation Metrics . . . 27

2.4.3 Implementation Details . . . 28

2.4.4 3D Object Classification . . . 29

2.4.5 3D Part Segmentation . . . 31

2.4.6 Ablation Study . . . 37

2.5 Discussion . . . 39

2.6 Summary . . . 40

Chapter 3 Deep Learning-based Analysis for Rotated Point Clouds 42

3.1 Introduction . . . 43

3.2 Literature Review . . . 45

3.2.1 Rotation Robust Feature Learning . . . 45

3.2.2 3D Point Cloud Registration . . . 45

3.2.3 Contrastive Learning with 3D Visual Correspondence . . . 47

3.3 Methods . . . 47

3.3.1 Rotation-Invariant Shape Encoding . . . 47

3.3.2 Aligned Feature Integration . . . 49

3.3.3 Deep Feature Registration 52
3.4 Experiments and Results 53
3.4.1 Datasets 53
3.4.2 Evaluation Metrics 54
3.4.3 Implementation Details 55
3.4.4 3D Object Classification 55
3.4.5 3D Part Segmentation 58
3.4.6 3D Shape Retrieval 58
3.4.7 Ablation Study 58
3.5 Discussion 63
3.6 Summary 65

Chapter 4 Deep Learning-based Modeling for 3D Human Mesh Texturing 66
4.1 Introduction 66
4.2 Literature Review 68
4.2.1 Score Distillation Sampling 68
4.2.2 Physically-Based Rendering 69
4.2.3 3D Shape and Texture Generation 70
4.3 Methods 72
4.3.1 Denoised Score Distillation 72
4.3.2 Geometry-aware Texture Generation 74
4.3.3 Semantic Zoom 75
4.4 Experiments and Results 75
4.4.1 Datasets 75
4.4.2 Evaluation Metrics 75
4.4.3 Implementation Details 76
4.4.4 Qualitative Results 76
4.4.5 Quantitative Results 78
4.4.6 Ablation Study 79
4.5 Discussion 83
4.6 Summary 84

Chapter 5 Deep Learning-based Analysis for Micro-expression Face Videos 86

5.1 Introduction 86

5.2 Literature Review 88

5.2.1 Generative Adversarial Network 88

5.2.2 Capsule Network 90

5.3 Methods 91

5.3.1 Micro-Expression Synthesis via Identity-aware Generator 91

5.3.2 Micro-Expression Recognition via Capsule-Enhanced Discriminator 94

5.3.3 Model Objective 95

5.4 Experiments and Results 97

5.4.1 Datasets 97

5.4.2 Evaluation Metrics 98

5.4.3 Implementation Details 98

5.4.4 Quantitative Analysis of MER 99

5.4.5 Qualitative Analysis of MES 101

5.4.6 Ablation Study 101

5.5 Discussion 105

5.5.1 Ethical Concerns 105

5.6 Summary 106

Chapter 6 Deep Learning-based Analysis for Text-Guided Face Video Generation 107

6.1 Introduction 107

6.2 Literature Review 109

6.2.1 Diffusion Models 109

6.2.2 Text-to-Video Generation 110

6.2.3 Multimodal Datasets 111

6.3 Methods 111

6.3.1 Data Collection & Processing 112

6.3.2 Data Annotation 113

6.3.3 Semi-auto Text Generation 115

6.3.4 MMVID-interp 116

6.4 Experiments and Results 117

6.4.1 Datasets 117

6.4.2 Evaluation Metrics 121

6.4.3 Implementation Details 122

6.4.4 Text-to-Video Generation 123

6.4.5 Benchmarks 124

6.4.6 Ablation Studies 128

6.5 Discussion 128

6.5.1 Ethical Concerns 129

6.6 Summary 130

Chapter 7 Conclusions and Future Work 131

7.1 Conclusions 131

7.2 Ethical Concerns 133

7.3 Future Work 134

Bibliography 136

Appendix i

Abbreviations i

List of Figures

1.1 Specific data Forms of three-dimensional data. We give detailed type of data examined this thesis: point clouds, human meshes, and face videos. 1

1.2 Visualization of data difference between medical and non-medical point clouds. Top part: 3D medical point clouds in IntrA [249] with complex and diverse topology. Bottom part: 3D general point clouds in ModelNet40 [237] with informative semantic structures and symmetry. 3

1.3 Errors introduced to model capability on shape classification, part segmentation, and model retrieval by unknown rotations on rotation-sensitive deep learning models such as PointNet [164]. 4

1.4 Data structure difference between 3D meshes and 3D point clouds, where 3D meshes present connectivity information between vertices and face information, while point clouds are sparse without connectivity information. Models (public domain) by Keenan Crane. 5

1.5 Dataset visualization of (a) Caesar [88] and (b) ScanDB [71]. Caesar contains 4300 registered meshes of 6890 vertices and 12K triangles. ScanDB contains 550 full body 3D scans of 114 subjects, each of which are scanned in at least 9 poses sampled randomly from 34 poses. 7

1.6 Overview of our methods for human avatar texturing, where we generate human textures given the rendered shape geometry and the input text again for high-quality and detailed textured human generation. 7

2.1 Model architecture of PointNet [164]. 17

2.2 Model architecture of PointNet++ [166], which effectively extracts geometric information from local shape regions in a hierarchical manner. 17

2.3 3D models of intracranial aneurysm segments from IntrA [249]. 18

2.4 Scaled Dot-Product Attention (left) and Multi-Head Attention (right) architectures taken from [210]. 19

2.5 Detailed architecture of our attention module in 3DMedPT, where KNN and FPS denote k-nearest neighbor and farthest point sampling operations [166], respectively. 22

2.6 Modified attention module with MGR, where SL denotes the self-loop and residual connection is applied to compensate the low-level information loss. All dotted components imply different designs with the original Lambda attention [15]. 24

2.7 The overall architecture of 3DMedPT for medical point cloud analysis. Numbers in black, blue, and orange indicate the point number, the feature dimension for classification, and the feature dimension for segmentation. LCA and RPE denote local context augmentation and relative positional embedding, respectively. 25

2.8 Qualitative comparisons on IntrA segmentation. Ground-truth samples are shown in the first column for reference. 32

2.9 Segmentation comparisons between ground-truth annotations and the outputs generated from the 3DMedPT on IntrA dataset. Good segmentation results are shown above the red line and some failed cases are shown below the red line. 33

2.10 Segmentation comparisons between ground-truth annotations and the outputs generated from the 3DMedPT on ShapeNetPart dataset. 35

2.11 **Left**: Comparison on different numbers of input points. **Right**: Comparison on different numbers of noisy points. 39

3.1 Frameworks of our design (left) and robust point cloud registration (right), where TI and RI are transformation invariance and rotation invariance, and $\mathbf{T}$ is the rigid transformation. The dotted line indicates the computation of $\mathbf{T}$ between reference frames. 44

3.2 The pipeline of a general registration method (i.e., iterative closest point (ICP) [16]). 46

3.3 Illustrations of (a) Intra-frame Aligned Self-attention and (b) Inter-frame Aligned Cross-attention modules. Note that we only present processes for computing $\mathbf{F}_{oa}$ in both modules. 50

3.4 Visualization of ScanObjectNN dataset. 54

3.5 Visualization of ShapeNetCore55 dataset. 54

3.6 Segmentation comparisons on ShapeNetPart, where ground-truth samples are shown for reference. Red dotted circles indicate obvious failures on certain classes, and purple circles denote the slight difference between our design and VN-DGCNN. 57

3.7 Visualization of semantic segmentation results on S3DIS area-5. The first row is the original inputs, the second row is the ground-truth samples and the last row is our predicted results. 60

3.8 Left: Results on Gaussian noise of zero mean and variant standard deviation values. Right: Results on different numbers of noisy points. 61

3.9 Network attention on PointNet++, RI-GCN and our model. 62

3.10 t-SNE of the aggregated $\mathbf{U}$ with z/SO(3) (**Left**) and SO(3)/SO(3) (**Right**). Clusters indicate good predictions in object classification. 63

4.1 Generated results of human avatars guided by textural descriptions using modified Fantasia3D [34], where human prior information (i.e., SMPL) is injected during the training pipeline. Input texts are shown below the rendered images. 71

4.2 Overview of our proposed model for text-to-human texture generation. Given an input text and a human mesh model, we generate a avatar texture to match the textual description. To achieve this, we propose Denoised Score Distillation with a negative pair of image and text prompts to guide the gradient direction for detail texture generation that is semantically aligned to the input text. We utilize a coordinate-based network with SV-BRDF to learn the material-related parameters (i.e., $\mathbf{k}_d$, r, and m) with depth maps for geometry-aware texturing. Finally, camera position is shifted with semantically adapted text input to refine the face region. 72

4.3 Generated texture maps for the diffuse term $\mathbf{k}_d$ and specular term $\mathbf{k}_s$. Texture dilation on UV islands is only applied on $\mathbf{k}_d$ to reduce texture seams. 74

4.4 Qualitative comparisons on RenderPeople [174] for textured human avatars against Latent-Paint [144], TEXTure [175], and Fantasia3D [34]. Our generation contains the best texture quality with high-frequency details and consistent with input textual descriptions. 77

4.5 Qualitative comparisons with DreamHuman [107]. As DreamHuman is not publicly available, we pick similar mesh models from RenderPeople [174] and download the results from the published paper. 78

4.6 Additional rendered results on RenderPeople [174] for textured human avatars guided by textual descriptions, where the corresponding input text is shown at the bottom of each sample. 80

4.7 Rendered results of textured Human avatars based on (a) SDS, (b) SDS with the depth map, (c) SDS with the depth map and negative prompts, and (d) DSD with the depth map. The dashed boxes indicate the mismatch between the texture and the textual descriptions, and the solid boxes denote human parts with low-quality textures. 81

4.8 Ablation study on different shading models. (a) Fully-proposed method with BRDF; (b) Fully-proposed method with Spherical Harmonic model (SH). 82

4.9 Importance of semantic zoom. The left image shows the generated avatar with semantic zoom, while the right image employs no semantic zoom. 82

4.10 Texture generation with challenging text prompts. 83

4.11 failure cases when using DSD for texture generation. Input textual descriptions for each sample are shown below the rendered images, where we show the front, side, and back views of the human avatar renderings. 84

5.1 The model pipeline for GAN [63]. 88

5.2 Detailed model architecture of the capsule network [220, 180]. 90

5.3 Architecture of the proposed ICE-GAN framework for MES and MER tasks, where G_{enc} and G_{dec} represent the encoder and decoder part of generator, and D denotes the capsule discriminator. 91

5.4 The encoder-decoder framework of our proposed generator G, where the detailed implementations of the graph construction and global reasoning are illustrated within the yellow box. 94

5.5 The capsule-enhanced discriminator D of our proposed method, where N_{prim} denotes the number of primary capsules, d_{prim}, d_{exp}, and d_{adv} represent the dimension of each *PrimaryCaps*, *ExpCaps*, and *AdvCaps*, respectively. 95

5.6 Extracted video samples from SMIC dataset [117]. 97

5.7 A demonstration of the frame sequence in a micro-expression from CASME II [246]. 97

5.8 A video sequence from SAMM dataset 5.8. 98

5.9 (a) Comparisons between real samples and generated samples by ICE-GAN for four different subjects. Neutral images are listed for reference. (b) Synthetic images X_{syn} generated by different model designs. 102

5.10 Norm2-based difference between X_{syn} and X_{on}. Green boxes indicate the subtle muscle movements associated with action units. 103

5.11 Failure cases when dealing with SMIC, CASME II, and SAMM datasets. 105

6.1 Diffusion model process cited from [80]. 109

6.2 **Pipeline of our dataset construction process.** The pipeline includes data collection & processing, data annotation, and semi-auto text generation. 112

6.3 **Dataset distribution comparison.** The distributions of appearance attributes, action attributes, and light directions. 114

6.4 **Dataset quality distribution.** The metrics used are BRISQUE [150] and VSFA [113] respectively. 117

6.5 **Text distribution.** CelebV-Text achieves better performance in both 4-gram and number words distribution. 119

6.6 **Qualitative results of facial text-to-video generation.** The generated samples are given texts describing (a) the static attribute and (b) dynamic attribute. 121

6.7 **Qualitative results on three facial text-video datasets.** Red and yellow regions indicate the missing of “bags under eyes" and the existence of “wavy hair" and “bags under eyes". 123

6.8 More sampled results from MMVID with input texts describing general appearances. 125

6.9 **Qualitative results of facial text-to-video generation.** The video samples are generated given texts describing dynamic emotion. 126

6.10 **Qualitative results of facial text-to-video generation on dynamic descriptions.** The video samples are generated given texts describing dynamic light directions. 127

List of Tables

2.1 Classification results of per-class accuracy and F1-score on healthy vessel segments (V.) and aneurysm segments (A.) with all input features. Results are averaged across all 5 folds. 30

2.2 Classification results on ModelNet40 with different input types and point numbers. 31

2.3 Segmentation results of each point-based network. V. and A. represent parent vessel segments and aneurysm segments. 34

2.4 Segmentation results of different methods on ShapeNetPart. 36

2.5 Positional embeddings in 3DMedPT evaluated on F1-score. RPE$_k$, RPE$_q$ and RPE$_v$ indicate relative positional embeddings at the key, query and value. 37

2.6 Model complexity of 3DMedPT for the IntrA classification, where model parameters are in the unit of millions and throughput is reported in examples per second. 38

2.7 Classification results on RetinalOCT with different methods. 40

3.1 Classification results on ModelNet40. All methods take raw points of 1024×3 as inputs. 56

3.2 Classification results on ScanObjectNN OBJ_BG. 56

3.3 Segmentation results on ShapeNetPart. The second best results are underlined. 57

3.4 Comparisons of SoTA methods on the 3D shape retrieval task. 59

3.5 Semantic segmentation results (mIoU) on S3DIS area-5. 59

3.6 Module analysis of AIT and loss functions. $\mathbf{F}^{g*}$ means encoding $\mathbf{F}^{g}$ via Intra-frame Aligned Self-attention. 59

3.7 Variance and Mean values of different model performances on ModelNet40 with z/SO(3). 64

3.8 Model complexity construction time for LRFs, and inference speed on ModelNet40 with z/SO(3), where [114] is considered without test time augmentation. 64

4.1 Quantitative comparisons of mean CLIP score between baseline models and ours for textured human avatars. Δ denotes the percentage by which our model outperforms the indicated method. 79

4.2 User study results of baseline models and ours. Δ denotes the percentage by which our model outperforms the indicated method. 79

5.1 MER results on the MEGC2019 benchmark and separate datasets in terms of UF1 and UAR with LOSO cross-database evaluation. 100

5.2 Comparisons with the latest graph-based and attention-based approaches in terms of F1-score with LOSO single-database evaluation. 100

5.3 Experimental analyses of G and GRM designs on the full dataset split in a subject-wise manner. SC, SE, and GR represent the skip connection, squeeze-and-excitation, and graph reasoning. 103

5.4 Ablation studies on the discriminator design. D_{CNN} and D_{CE} denote CNN-based and capsule-enhanced discriminators. 105

6.1 **In-the-wild face video dataset comparison.** The symbol "#" indicates the number. The abbreviations "Res.", "Dura.", "App.", "Cond.", "Act.", "Emo.", and "Dir." stand for Resolution, Duration, Appearance, Condition, Action, Emotion, and Direction, respectively. The "half checkmark" denotes that CelebV-HQ consists of action attributes with no timestamp. 118

6.2 **Multimodal retrieval results.** Clip2Video [60] is leveraged to measure the text-video relevance via retrieval experiments. Bold values indicate the best results, underlined ones indicate the second best. 118

6.3 **Number of unique POS tags.** The numbers of unique POS tags for MM-Vox, CelebV-HQ, and CelebV-Text. 120

6.4 **Benchmark of text-to-video generation on different datasets.** ↓ means a lower value is better and ↑ means the opposite. 129

CHAPTER 1

Introduction

Three-dimensional (3D) data represents information that exists in three dimensions or axes, where 3D data take into account not only length and width but also depth or height, often used to describe objects or environments in the real world. 3D data plays an increasingly crucial role in numerous domains, including autonomous driving, computer graphics, and medical imaging [64, 249, 39].

Traditional forms of 3D data include 3D mesh models and 3D point clouds [90, 203]. However, for a thorough investigation of 3D data, we follow [90] which considers a video as 3D data, where time is considered as the third dimension along with spatial dimensions (width and height) of the frames. In this thesis, we consider three forms of 3D data: 3D point clouds, 3D human meshes, and face videos. Despite the substantial potential of these data types, their unique characteristics pose significant challenges. In this section, we give brief introductions and challenges, respectively.

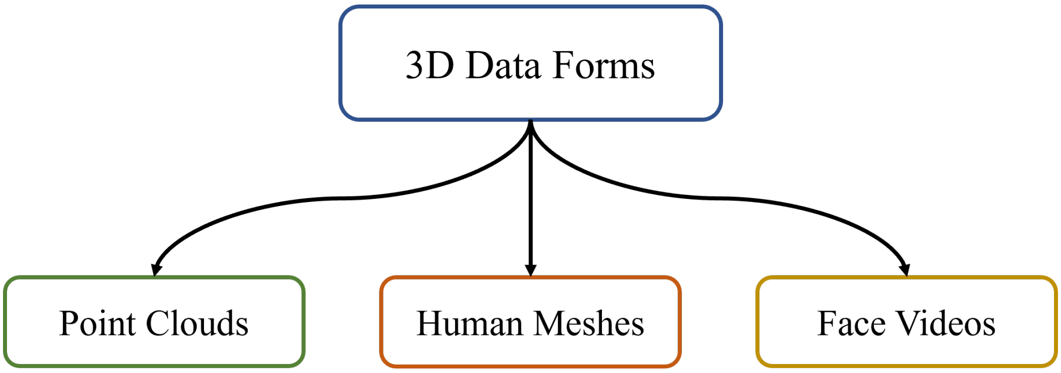


FIGURE 1.1. Specific data Forms of three-dimensional data. We give detailed type of data examined this thesis: point clouds, human meshes, and face videos.

1.1 3D Point Clouds

Point cloud data, due to its unstructured and unordered nature, requires specialized handling. Key challenges in point cloud analysis include the handling of noise, the need for computational efficiency, and the extraction of semantic understanding from complex 3D structures. This is particularly evident in tasks such as classification and segmentation [67]. The advent of deep learning methods specifically for 3D methods [164, 166, 231, 86] has provided a pathway to address these challenges. For example, PointNet [164] directly feeds the 3D coordinates of points into the convolution-based network and presents a better performance on shape classification than project-based methods [165, 142, 237], which regularize the structure of 3D points by voxelization or projection, leading to the loss of intrinsic geometric information.

1.1.1 Challenges

Although recent advancements in point cloud analysis have demonstrated great success on the shape classification and segmentation tasks, there still remains challenges:

- Existing models can perform well on general 3D datasets, but they could fail on medical point clouds due to the domain gap and the irregular topology introduced in medical data.
- 3D objects are normally rotated and their poses are unknown in real scenarios, which can largely impact the deep learning models that are sensitive to rotations.

In the following, we give more detailed analyses of the abovementioned challenges. We give our solutions for dealing with medical point clouds in Chapter 2 and rotated point clouds in Chapter 3.

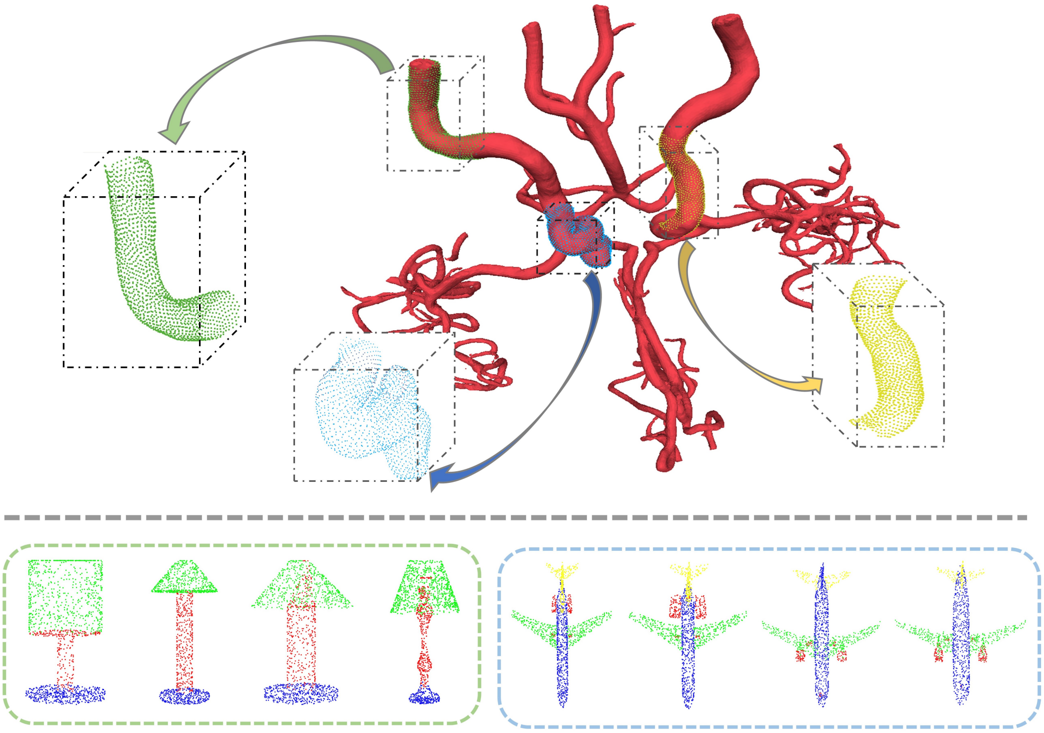


FIGURE 1.2. Visualization of data difference between medical and non-medical point clouds. Top part: 3D medical point clouds in IntrA [249] with complex and diverse topology. Bottom part: 3D general point clouds in ModelNet40 [237] with informative semantic structures and symmetry.

1.1.2 Medical Point Clouds

Medical point cloud data has become increasingly critical in numerous applications related to healthcare and medical research, spanning from surgical planning to diagnostic analysis, and from patient-specific implant design to the development of prosthetics [249]. Accurate pathological segments of medical data are important for disease diagnosis and treatment. However, 3D medical data can contain incomplete pathological structures, which are hard to distinguish from healthy parts within an object.

For point clouds of general objects (bottom part of Figure 1.2), they usually present a similar pattern for non-medical datasets. In contrast, as shown in the top part of Figure 1.2, medical point clouds of the same class have more diverse and complex shapes in geometry and topology, which introduce difficulties for object classification or segmentation. Insufficient samples of medical data also make it difficult to learn distinctive shape descriptors. Hence, it is essential to design an effective deep learning method which can demonstrate good performances on medical data and also generalize well on non-medical data.

1.1.3 Rotated Point Clouds

A particular challenge in the analysis of 3D point clouds is rotation invariance. In many real-world applications, the same object or scene can be observed from different angles or orientations, resulting in rotated versions of the same point cloud. This can significantly affect the performance of learning algorithms that are sensitive to input orientation, leading to inconsistent and unreliable results. As shown in Figure 1.3, PointNet trained with 3D objects in the canonical pose fails to recognize the same object seen from a random angle.

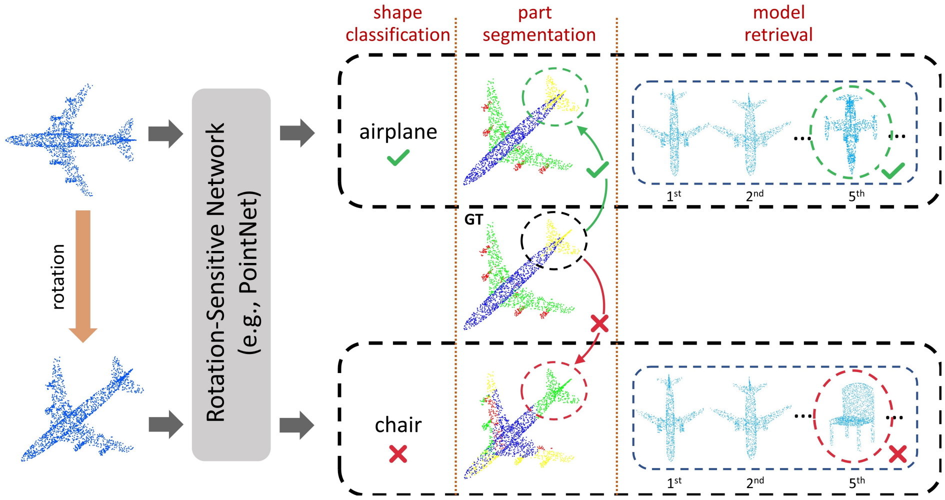


FIGURE 1.3. Errors introduced to model capability on shape classification, part segmentation, and model retrieval by unknown rotations on rotation-sensitive deep learning models such as PointNet [164].

To embed rotation invariance property to existing methods for point cloud analysis, a straightforward way is augmenting training data with massive rotated point cloud samples which, however, requires a large memory capacity and exhibits limited generalization ability to unseen data [103]. Recently, some approaches have been proposed to embed the network with rotation invariance. Works such as [59, 43, 253] project 3D raw points to meshes and process data with spherical convolutions. However, they are still sensitive to rotations due to point cloud projections and lack equivariant non-linearity. Therefore, others propose to replace raw

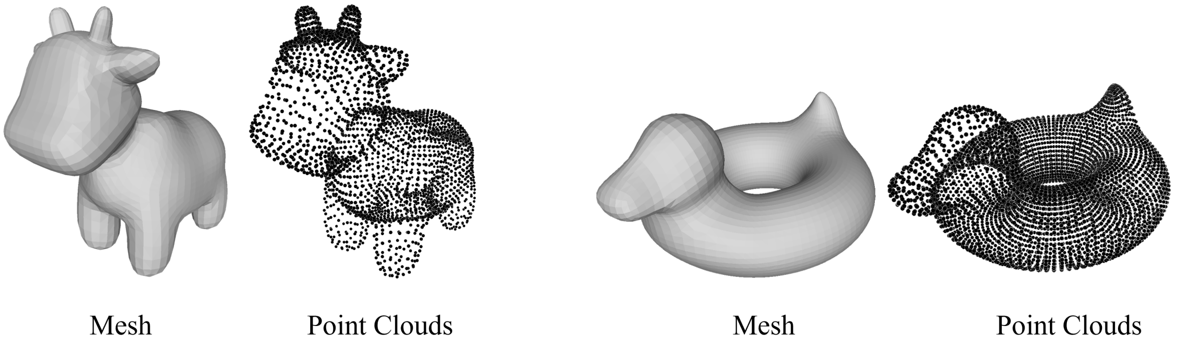


FIGURE 1.4. Data structure difference between 3D meshes and 3D point clouds, where 3D meshes present connectivity information between vertices and face information, while point clouds are sparse without connectivity information. Models (public domain) by Keenan Crane.

points with rotation invariant features as model inputs. RI-GCN [103] designs features based on local reference frames (LRFs), i.e., a local coordinate system. Although local geometric cues between points are preserved, the loss of global information could hinder the model performance. Further works such as [116, 256] utilize features from both local and global domains, which however did not investigate how to fully integrate knowledge from these two domains. Therefore, developing an effective method for rotated point cloud processing is urgent.

1.2 3D Human Meshes

As another primary representation of 3D data, a 3D mesh is a collection of vertices, edges, and faces that define the shape of a 3D object. The vertices are points in 3D space, edges are lines connecting pairs of vertices, and faces are polygons formed by three or more connected edges. Unlike point clouds that are sparse and have no connectivity information as shown in Figure 1.4, 3D meshes contain connectivity information, which means that each vertex, edge, and face knows what it is connected to. This information can help in identifying the structure of the object. Moreover, 3D meshes include topological consistency, enabling operations such as smoothing, simplification, and parametrization can be performed on them, which is useful in applications like computer graphics and geometric modeling.

In this thesis, we focus on human meshes, as opposed to general meshes, which offers several important advantages and opportunities for research and application, largely due to the significance of human models in many domains. For example, detailed 3D human mesh models enable sophisticated human-computer interaction [211], and in the realm of computer graphics, human meshes can lead to more accurate and natural-looking character models and animations [213].

1.2.1 Challenges

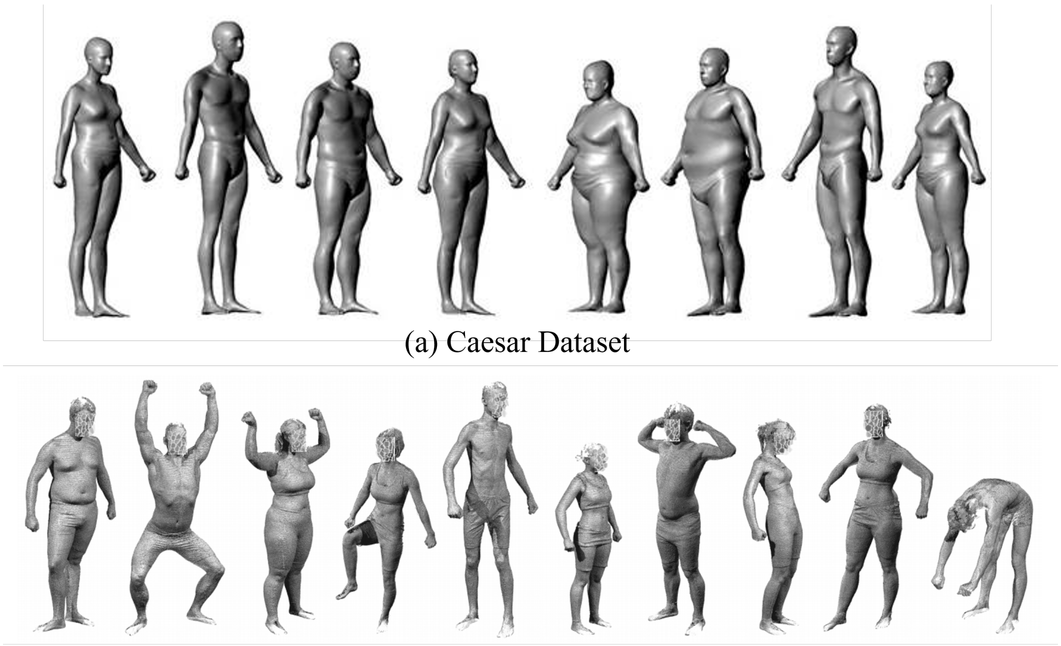
Here we present the challenges associated to human mesh models, so that we would like to address to make the human mesh modeling more practical and useful in real world scenarios:

- Most human mesh models contain no textures (see Figure 1.5), such as ScanDB [71] and Caesar [88], while texture acts an important role as human perception is particularly sensitive to the texture. Designing a high-quality and detailed texture is also a challenging task.

In the following, we give more detailed analysis of the challenges about 3D human mesh modeling as well as texture generation. We present our solutions in Chapter 4.

1.2.2 3D Human Texture Generation

The domain of 3D human texture estimation has seen remarkable advancements in recent years. Some approaches [4, 5, 6] address this challenge by employing multi-view images as input, from which they synthesize 3D avatar textures by merging textures derived from varying perspectives. Alternatively, other methods [7, 271, 163] reconstruct human textures using monocular images. However, these works typically require 3D supervision obtained through 3D scanning, a process that is both labor-intensive and costly. Recent methods address this problem without 3D labels [244, 62], which are more applicable for real-world usage. However, they still lack control over the texture generation process as the final results are completely determined by the input image.


(a) Caesar Dataset

(b) ScanDB Dataset

FIGURE 1.5. Dataset visualization of (a) Caesar [88] and (b) ScanDB [71]. Caesar contains 4300 registered meshes of 6890 vertices and 12K triangles. ScanDB contains 550 full body 3D scans of 114 subjects, each of which are scanned in at least 9 poses sampled randomly from 34 poses.

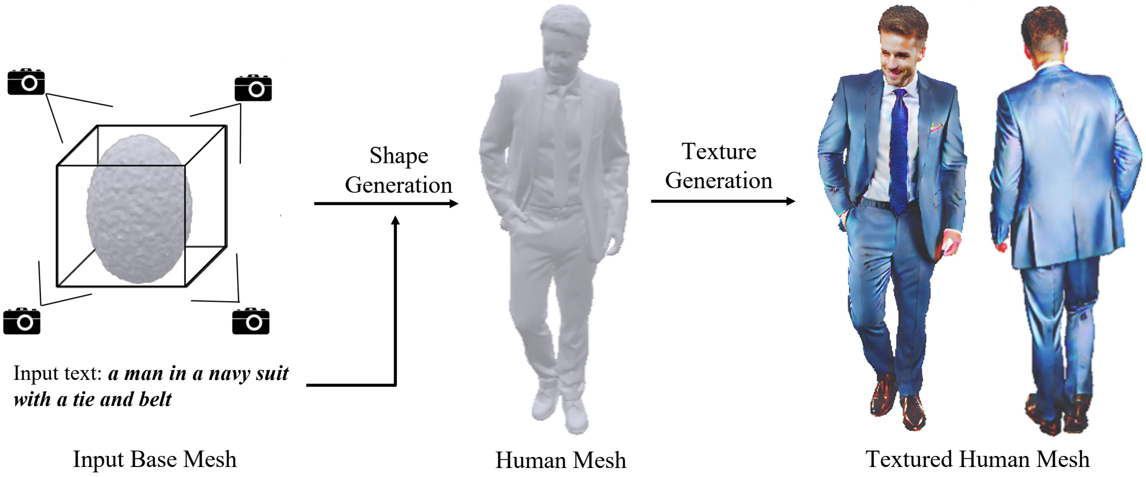


FIGURE 1.6. Overview of our methods for human avatar texturing, where we generate human textures given the rendered shape geometry and the input text again for high-quality and detailed textured human generation.

To generate textures to a high degree of freedom, as shown in Figure 1.6, we consider textual descriptions a direct and convinient way for 3D avatar texture synthesis, where rendered human avatar contains high-quality and detailed textures semantically aligned with input texts. Due to the success of text-to-image generation which leverages the diffusion generative model [159, 80] and Score Distillation Sampling (SDS) [162], the parametric human image generator using the 2D diffusion model as a prior can be optimized. Hence, zero-shot 3D mesh texturing guided by textual descriptions are made possible. However, generating a texture map for a 3D human avatar in a zero-shot manner is challenging due to two reasons. First, SDS tends to guide the model to converge towards a specific mode, resulting in over-smoothed or blurry human body parts, or rendered human image not being faithfully semantic with the input texts. Second, the generated texture maps are often semantically unaligned with the human mesh surface or missing textures for complex garment details.

Generating a 3D human texture is normally posed as an image inpainting task (i.e., TEX-Ture [175] and Text2tex [31]), which utilize the prior knowledge of two Stable Diffusion models [176] for incremental texture inpainting. However, the alignment of rendered images under different viewpoints could not be guaranteed for the human texture generation task. Moreover, due to the lack of knowledge of the human meshes, complex garment details such as clothes wrinkles may not be textured, or textures of different clothing items can be hard to distinguish and merged together. On the other hand, methods such as Latent-Paint [144] or Fantasia3D [34] capitalize on SDS by distilling the prior information of a pre-trained diffusion model for texture generation via differentiable rendering. Although the meshes can produce textures aligned to human meshes, they also produce unfaithfully semantic textures with input texts. Moreover, despite the usefulness of SDS, one of the primary issues associated with SDS is its gradient direction might converge to a specific mode, which causes non-detailed and over-smoothed body parts.

1.3 Face Videos

When we traditionally consider a video, we consider it as a 2D sequence of images, while it is more appropriate to consider video as 3D data, where two dimensions represent the spatial information (height and width) of each frame, and the third dimension is time, representing the sequence of frames. This spatial-temporal nature of videos makes them a form of 3D data [90].

In this thesis, we study take face videos as our main research object. Studying face videos in the deep learning domain holds great potential and significance due to two reasons: 1) Emotion recognition: Human facial expressions are a rich source of non-verbal communication that carry essential information of the emotional state of a person. Deep learning models trained on face videos can identify subtle emotion changes (i.e., micro-facial expression recognition) and help in contributing to fields such as human-computer interaction, psychology, and health care [202]; 2) Synthesis and editing: Deep learning models can generate or modify face videos, which is crucial for research purposes, movie production, animation, or game development.

1.3.1 Challenges

Here we present existing challenges when processing face video data:

- General deep learning-based models can recognize and generate fake videos with macro-expressions. However, developing a model for micro-expressions recognition and synthesis can be more challenging than macro-expressions, because micro-expressions display unconscious feelings with low facial expression density that is hard to be accurately perceived by deep learning models.
- Although video generation models conditioned on texts are flourishing in video generation and editing. However, face-centric text-to-video generation remains a challenge due to the lack of a suitable dataset containing high-quality videos and highly relevant texts, resulting in generated face videos of low-quality or low control on the generated face videos using textual descriptions.

In the following, we give more detailed analysis of the challenges related to video-based micro-expressions recognition tasks and text-guided face video generation task. Detailed solutions are to be presented in Chapters 5 and 6.

1.3.2 Micro-Expression Face Video

Video-based micro-expressions (MEs) contain subtle facial expression change that can be hardly perceived by untrained observers, making it a challenging task. It has attracted an increasing number of researchers into the study of micro-expression recognition (MER) due to its practical applications in lie detection and disease diagnosis [156, 143, 235]. There have already been many successful works proposed for general expression recognition (also known as macro-expression recognition), but the domain of MER is poorly-explored mainly due to 1) existing methods cannot handle facial expressions with low density and 2) lack of large-scale ME datasets to support extensive MER studies.

Hand-engineered methods such as Facial Action Coding System (FACS) [44] are applied to recognize facial expressions. FACS focuses on muscles that produce facial expressions and measures the movement with the help of action units (AUs). Two systems are further developed: the Micro Expression Training Tool (METT) and Subtle Expression Training Tool (SETT) [96]. However, the best classification accuracy achieved by METT/SETT is still not satisfactory because the result is heavily affected by humans, making the detection unconvincing and unstable. Local binary pattern (LBP) and local quantized pattern (LQP) are later developed [143], and LBP with three orthogonal planes (LBP-TOP [265]) has shown superiority in processing facial images. However, these geometry-based methods rely heavily on the proposed images and can be easily affected by global changes.

With the development of deep learning technologies, works have been proposed based on data-driven approaches utilizing the convolutional neural network (CNN) for micro-expression recognition [101, 130, 209]. However, CNN-based methods are invariant to translations and are unable to encode positional relations and and orientation information of different facial entities. Furthermore, due to the data scale limitation of facial expression samples, data-driven model performance can be heavily constrained.

1.3.3 Text-guided Face Video Generation

Text-driven general video generation has recently garnered significant attention in the field of computer vision and computer graphics. By using text as input, video content can be generated and controlled, inspiring numerous applications in both academia and industry [119, 13, 171, 151].

However, text-to-video generation still faces many challenges, particularly in the face-centric scenario where generated video frames often have weak relevance to input texts [12, 272, 141, 3]. We believe that one of the main issues is the absence of a well-suited facial text-video dataset containing high-quality video samples and text descriptions of various attributes highly relevant to videos.

Constructing a high-quality facial text-video dataset poses several challenges, mainly in three aspects: 1) *Data collection.* The quality and quantity of video samples largely determine the quality of generated videos [230, 155, 41, 178]. However, obtaining such a large-scale dataset with high-quality samples while maintaining a natural distribution and smooth video motion is challenging. 2) *Data annotation.* The relevance of text-video pairs needs to be ensured. This requires a comprehensive coverage of text for describing the content and motion appearing in the video, such as light conditions and head movements. 3) *Text generation.* Producing diverse and natural texts are non-trivial. Manual text generation is expensive and not scalable. While auto-text generation is easily extensible, it is limited in naturalness.

Besides the effect brought by face-centric dataset, existing methods for conditional video generation often generate face video of low quality [68, 131, 119]. Most of the methods apply conditional GAN variations to video data. Despite certain achievements, they face the following constraints: (1) The generator network utilizes 3D transposed convolution layers, resulting in the synthesized videos of fixed length only. (2) Their model inputs are designed to take videos of low resolution, displaying results solely at a 64×64 resolution. (3) Encoded video and text features are simply concatenated for further processing, which could cause information loss and could be difficult to extract rich text-video relations.

Hence, developing a new dataset including paired texts and face videos and designing an effective methods for text-conditioned face video generation is quite urgent.

1.4 Thesis Contributions and Organization

In this thesis, we take further research to tackle the challenges in all three mentioned 3D data types: point clouds in Section 1.1.1, human meshes 1.2.1, and face videos 1.3.1. Based on different learning objectives, the main body of this thesis is divided into four parts to address the challenges observed in the abovementioned three kinds of data types.

In Chapter 2, we propose a deep learning-based model to deal with medical point clouds, where we design a transformer-based architecture to process the irregular data topology and the data domain gap imposed by medical point clouds as mentioned in Section 1.1.1. Our method is constructed with a novel transformer-based model, specifically tailored to analyze intricate structures in medical point cloud data, such as those found in organ-specific scans or disease-specific anomalies. Our attention module improves existing techniques by augmenting contextual information and summarizing local responses at the query, thereby capturing a comprehensive picture of both local context and global content feature interactions. Recognizing that insufficient training samples in medical data can lead to poor feature learning, we introduce two modules to address this issue: 1) We utilize position embeddings to accurately understand local geometric structures; 2) We deploy multiple graph-based reasoning blocks to examine the global knowledge propagation over feature channels to enrich feature representations. The content of this chapter is based on my previous work [7] in the publication list.

In Chapter 3, we propose a novel method to extract rotation invariant features from randomly posed point cloud samples, such that we could tackle the problem mentioned in Section 1.1.1. We consider rotation invariance as a variant of point cloud registration task and proposes an effective framework for rotation invariance learning by firstly encoding low-level rotation invariant shape descriptors extracted from local patches and global topology. We then introduce a novel position encoding module that leverages angle differences between different reference frames to align the features learned from local and global ranges. The aligned features are integrated by a customized attention mechanism which is embedded in our transformer architecture design, with a final feature integration module to ensure the rotation invariance by using a contrastive loss function. The content of this chapter is based on our previous publication [3] in the publication list.

We address the problem associated with 3D human meshes on texturing in Chapter 4, where we generate human textures conditioned on textural descriptions. Due to Score Distillation Sampling (SDS) introduced in [162] along with prior information of the pre-trained text-to-image diffusion model [176], zero-shot 3D content generation conditioned on textual descriptions are made possible. Inspired by physically based rendering (PBR) and the Bidirectional Reflectance Distribution Function (BRDF) [98], we propose our method to synthesize human avatar textures given the mesh model and the textual descriptions in a zero-shot manner. We design a novel score function based on a pre-trained depth-to-image diffusion model [80, 176], which enables the generation of high-quality rendered human images. In addition, we employ a coordinate-based network (i.e., Instant-NGP [154]) to estimate the BRDF model for surface materials prediction, which ensures the generated texture is semantically aligned with human mesh surfaces. Our rendering model is made differential with BRDF for photorealistic human texturing. The content of this part is based on [1] in our publication list.

In Chapter 5, we tackle the challenge of generating and classifying micro-expression face videos. Specifically, we present a method for video-based micro-expression recognition and synthesis problem. We first use an encoder-decoder network to learn the original data distribution with identity information, where a Graph Reasoning Module (GRM) is applied between the encoder and decoder networks for effective feature learning. We then use a Generative Adversarial Network (GAN) [63] to synthesize new face samples. By combining the identity-aware embedding and the class label, our GAN can generated new face samples with controlled micro-expressions in an identity-aware manner. Lastly, we propose an approach based on the Capsule network [180] to encode the translation and relative position information between different facial entities. The Capsule network is used for two purposes: 1) It enables the micro-expression recognition with a classifier; 2) It acts as an essential module for GAN as a discriminator to distinguish whether the input data is fake or real. This chapter is developed based on our publication [6] in the publication list.

In Chapter 6, we introduce a large-scale facial text-to-video dataset and a text-guided face video generation method for realistic face video generation. First, we introduce the detailed step of data collection, analysis and an automatic generation pipeline for textual descriptions given each face video. Second, we propose a generation framework conditioned on textual descriptions, where we utilize quantized representations for videos [176], and employ a bidirectional transformer [210]. The transformer takes both text and video tokens as inputs to generate discrete video tokens. To enhance video quality and ensure consistency, we introduce a novel video token, which is trained through a self-learning mechanism. We also introduce text interpolation on temporal domain to improve the alignment between textual descriptions and generation videos. The methods, results, and discussions in this chapter are based on our published work [2] in our publication list.

In Chapter 7, we draw our conclusions from the research presented in this thesis, encapsulating the key findings, their implications, and the value of our work. Furthermore, recognizing that the field of 3D data analysis continues to evolve, we also outline future research directions.

CHAPTER 2

Deep Learning-based Analysis for Medical Point Clouds

The inherent complexity of point cloud data, especially for its unstructured and unordered nature, necessitates the use of specialized tools and techniques for its effective analysis. Challenges related to point cloud analysis include the mitigation of noise, the need for efficient computational strategies, and the extraction of salient semantic insights from complex 3D geometries, etc. The advent of deep learning methods[164, 166] specifically designed for handling 3D data provides a promising strategy to tackle these obstacles. In this chapter, we aim to address the issue proposed in medical point clouds. Specially, we introduce an attention-based network with graph neural networks to examine the irregular topology and complex shape of medical point clouds. Our method demonstrates superior results on medical point cloud classification and segmentation tasks, and also perform well on general point cloud datasets when compared to existing SoTA methods.

2.1 Introduction

Inspired by the success of Transformer in both natural language processing and computer vision domains [55, 25, 53, 210], we propose an attention-based model for the medical point cloud processing. Self-attention is inherently order-invariant because attentional weights between the query and key remain the same if the input order is changed, which introduces permutation invariance, making it suitable for handling 3D point clouds. Moreover, attention can model long-range dependencies and learn expressive features. Based on these perspectives, we decide to analyze the medical point clouds based on attention layers. To address the issue of the irregular layout intrinsic to medical data, we augment the input feature with contexts from local neighborhood before each attention module, making our network context-aware. Meanwhile, the contextual information at query is further summarized via convolution to generate holistic geometric features. Moreover, due to the small-scale medical dataset compared to general ones, we learn diverse positional embeddings at query, key and value, and we propose Multi-Graph Reasoning to concurrently establish multiple channel graphs over the same feature nodes, with variant learnable adjacency matrices to enrich feature expressions.

Contributions of this work for addressing 3D medical point clouds are summarized as follows:

(1) We propose a Transformer-based network, namely 3DMedPT, to capture local context interactions via attention, and introduce convolution into Transformer to summarize global point features to obtain global content exchange within medical point clouds.
(2) We apply positional embeddings to address the irregular geometries of medical data.
(3) We design Multi-Graph Reasoning which captures global relations among feature channels to enrich the representational power of medical features at deep layers.
(4) Our model ranks the 1st in both classification and segmentation tasks in IntrA benchmark and reveals good generalization ability on ModelNet40 and ShapeNetPart.

2.2 Literature Review

2.2.1 3D General Point Clouds

Existing methods propose 3D deep learning models for general point clouds. Methods such as [164, 166, 128] apply point-wise Multi-Layer Perceptrons (MLPs) to analyze the 3D points. PointNet [164], as shown in Figure 2.1, is a pioneer and fundamental that processes 3D general point clouds with shared MLPs, where the geometry information is further aggregated via pooling layers. However, PointNet is unable to extract point relations within local geometry, which degrades the model performance on segmentation. Later work such as PointNet++ [166] as shown in Figure 2.2 investigates the underlying geometry by grouping information from local geometry. Other works [118, 231, 199, 218] handle point clouds

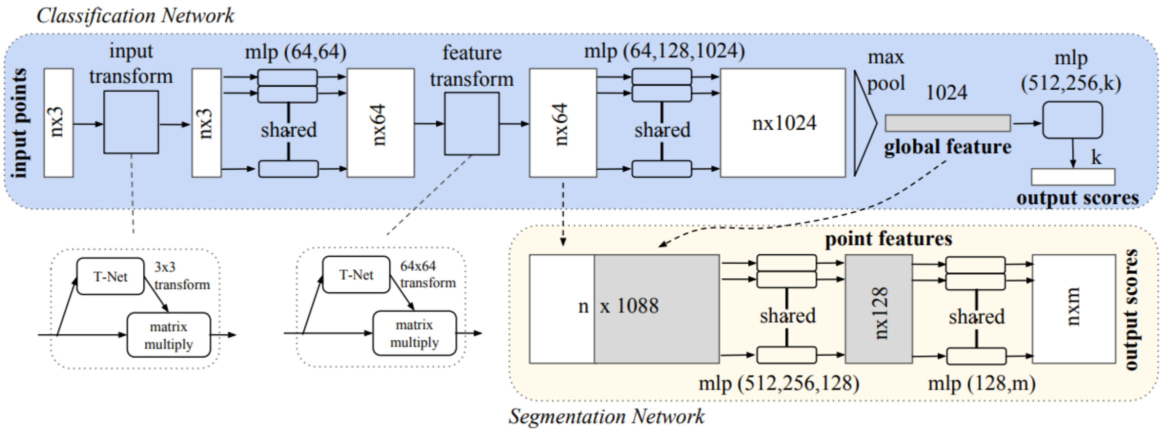


FIGURE 2.1. Model architecture of PointNet [164].

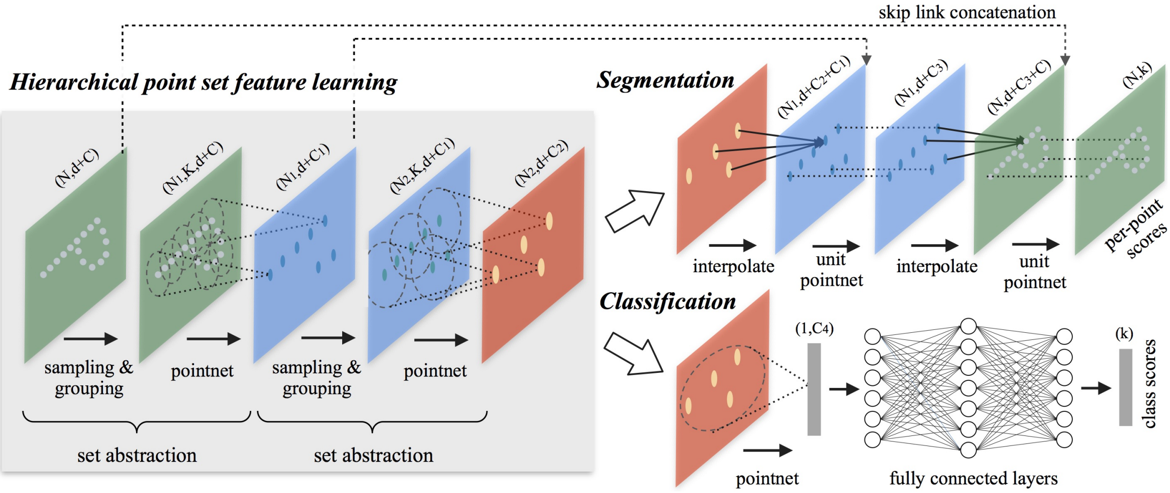


FIGURE 2.2. Model architecture of PointNet++ [166], which effectively extracts geometric information from local shape regions in a hierarchical manner.

with continuous convolutional kernels, while others [223, 243, 217] utilize graphs and graph convolution for point-wise feature encoding to learn local 3D structures. However, these methods all adopt MLPs to process point features, which constrains the model ability in capturing more expressive shape information.

Later works such as [200, 197] are purely based on MLPs, which explore the strength of MLPs for processing 3D point clouds. MLP-mixer [200] and Synthesizer [197] are two pioneers in this stream that are solely built based on MLPs. PointMixer [38] is the first work that uses MLP-like structures for point cloud analysis, where the query point feature is aggregated and

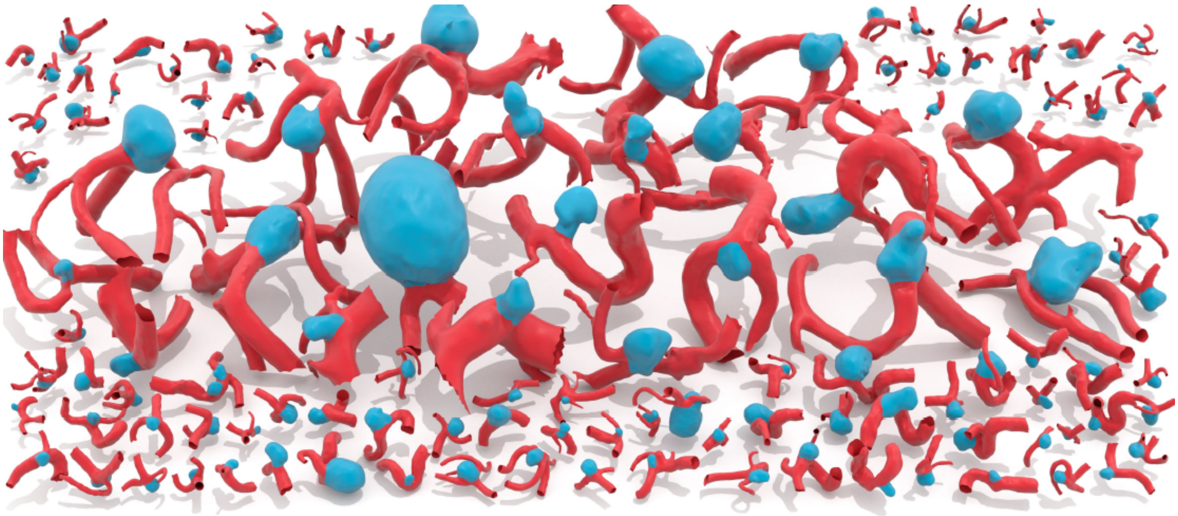


FIGURE 2.3. 3D models of intracranial aneurysm segments from IntrA [249].

updated from different dimensions. However, PointMixer cannot perform well on the shape classification task.

2.2.2 3D Medical Point Clouds

Due to the Advent of a publicly available 3D intracranial aneurysm dataset (shown in Figure 2.3), IntrA [249], the investigations of points-based and mesh-based classification and segmentation models using deep learning methods are made available. Different from 3D general point clouds, the data collection in medical domain is inefficient and dataset size is small. Although objects of arbitrary shapes can reveal critical information rather than simple Euclidean geometry, the analysis and modeling of these objects of complex shapes need special treatment. For medical point cloud processing, works such as [20, 73] directly borrow PointNet/PointNet++ for vessel labeling and aneurysm segmentation, and [11] modifies PointNet with a hand-engineered geometry learning algorithm. Nonetheless, these works are still MLP-based, and the issue of complex and incomplete geometry of medical data cannot be well addressed. In contrast, we process medical point clouds based on Transformer network in an effective manner.

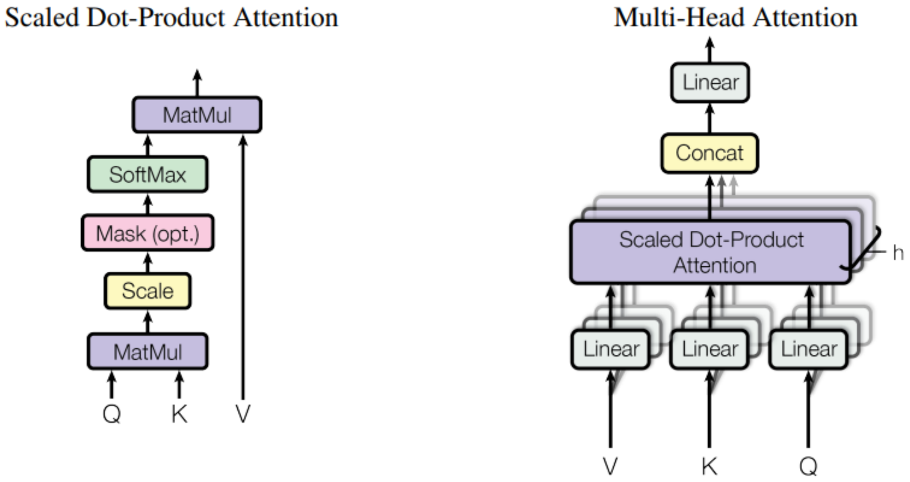


FIGURE 2.4. Scaled Dot-Product Attention (left) and Multi-Head Attention (right) architectures taken from [210].

2.2.3 Transformer

Transformer [210] has shown great success in natural language processing (NLP) and machine translation tasks [53, 250], which also encourages popular applications for 2D image processing [55, 25, 225, 132]. In the following, we review the implementation of the attention mechanism introduced in Transformer within the context of 3D point clouds. The detailed architecture for the attention mechanism is shown in Figure 2.4.

Suppose we are given an unordered set of N point features $\mathbf{P} \in \mathbb{R}^{N \times C}$ with C feature channels. Three matrices can be learned through linear mappings:

$$\mathbf{Q}, \mathbf{K}, \mathbf{V} = \mathbf{P}\mathbf{W}_q, \mathbf{P}\mathbf{W}_k, \mathbf{P}\mathbf{W}_v, \tag{2.1}$$

where $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{N \times C_k}$ and $\mathbf{V} \in \mathbb{R}^{N \times C_v}$ have output feature dimensions of C_k and C_v, respectively, and W_q, $W_k \in \mathbb{R}^{C \times C_k}$, and $W_v \in \mathbb{R}^{C \times C_v}$ are learnable weights. Following the terms defined in [210], $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are noted as query, key, and value matrices, respectively, and the self-attention operation is formulated as follows:

$$\mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{C_k}}\right)\mathbf{V}. \tag{2.2}$$

As shown in Eq. 2.2, global attention weights calculated from $\mathbf{Q}$ and $\mathbf{K}$ have a time complexity $\mathcal{O}(N^2C_v)$ and space complexity $\mathcal{O}(N^2 + NC_k + NC_v)$, which increase quadratically when N increases and consumes much computational resources.

Due to the quadratic computational cost of attention matrix implementation, a large amount of memory space is needed to deal with even short input sequences. A number of methods have been devoted to designing efficient attention implementations. Works such as [179, 37, 105] use sparse matrices with strict constraints for efficient attention computation. Other works [40, 15, 99, 216] employ kernel factorization or matrix factorization to reduce the computational overhead.

Here we present the mathematical expressions of Lambda attention:

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{Q}\big(\text{softmax}(\mathbf{K})^\top \mathbf{V}\big), \tag{2.3}$$

where *keys* are normalized through the softmax function, and $\text{softmax}(\mathbf{K})^\top \mathbf{V} \in \mathbb{R}^{C_k \times C_v}$ is termed as content lambda [15], where each query can interact with the content lambda in a linear form. Therefore, the time and space complexities are $\mathcal{O}(NC_kC_v)$ and $\mathcal{O}(NC_k + NC_v + C_kC_v)$, respectively, where the computational cost could be largely reduced when $C_k \ll N$. For efficient computation, we choose Lambda attention as our baseline model for medical point cloud processing. Lambda attention [15] reinterprets the attention as similarity kernels so that linear computations of attention are achieved, and axial-attention [216] decomposes 2D attention matrix into two 1D matrices along the width and height dimensions. Recently, the Swin Transformer [132] uses shifted windows to save computational cost. However, different from the initial purpose of Lambda attention in the 2D domain, we modify input features by augmenting local contexts and relative positional bias to address the complex topology and irregular geometries of 3D medical point clouds.

Most recently, several Transformer-based networks have been proposed for general 3D point cloud analysis. PCT [66] utilizes input embedding and offset attention to improve the network behavior. Point Transformer [266] modifies vector attention with relative positional embeddings to construct hierarchical attention layers for point cloud analysis. In this work, we propose an attention-based model that specifically works for medical point clouds with good generalization ability on non-medical datasets as well.

Convolutions have also been introduced into the Transformer block to utilize the effectiveness of CNNs, either by replacing multi-head attentions with convolution [228] or adding more convolutional layers to capture local correlations [232, 133]. Different from all the previous works, we propose convolution operation (i.e., EdgeConv [223]) solely on query features to summarize local responses from unordered 3D points to generate global geometric representations, of which the purpose is totally opposite to [133].

2.2.4 Graph-based Reasoning

We first review the idea of Graph Convolutional Network (GCN). Given the edge $\mathcal{E}$ and nodes $\mathcal{V}$ in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the goal is to learn a function of signals/features on the graph with (1) a feature matrix $X \in \mathbb{R}^{N \times D}$, where N is the number of nodes and D is the feature dimension, and (2) an adjacency matrix $A \in \mathbb{R}^{N \times N}$ that describes the relation of nodes in the graph. Using the input feature and the adjacency matrix, the aim is to learn a node-level output $Z \in \mathbb{R}^{N \times F}$, where F is the output feature dimension and graph-level output can be summarized via pooling [57]. Then we can use the non-linear function to represent each layer:

$$H^{(l+1)} = f(H^{(l)}, A), \tag{2.4}$$

where $H^{(0)} = X$ and $H^{(L)} = Z$, with L being the number of layers. Then the layer-wise propagation function is defined as:

$$f(H^{(l)}, A) = \sigma\left(AH^{(l)}W^{(l)}\right), \tag{2.5}$$

where $W^{(l)}$ is the weight matrix for l-th layer, and σ is a non-linear activation function (e.g., ReLU). For details about GCN, please read [104].

Recently, graph convolutions [104] have been adopted to capture relations between objects. Graph-based reasoning has been adopted to achieve global relation reasoning over 2D image graphs [261]. However, general graph convolutions can only be safely applied when node connections are known, reasoning over point clouds is challenging since no link information between nodes is present. Super-point graph [110] constructs graphs over 3D points with huge computational costs. Inspired by [140], we construct graphs on feature channels to avoid dealing with a large number of points. Moreover, to address the insufficient training samples in medical domain, we propose to construct multiple reasoning graphs over the same point features in parallel with learnable adjacency matrices for enriched global information learning.

2.3 Methods

We firstly explain how to embed local contexts for unordered 3D point clouds via contextual information augmentation and how to summarize local responses at query. Relative positional embeddings are then proposed to handle local geometry of medical data. Finally, we propose Multi-Graph Reasoning (MGR) on feature channel domains to enrich the representations of learned features. Our model architecture is shown in Figure 2.5.

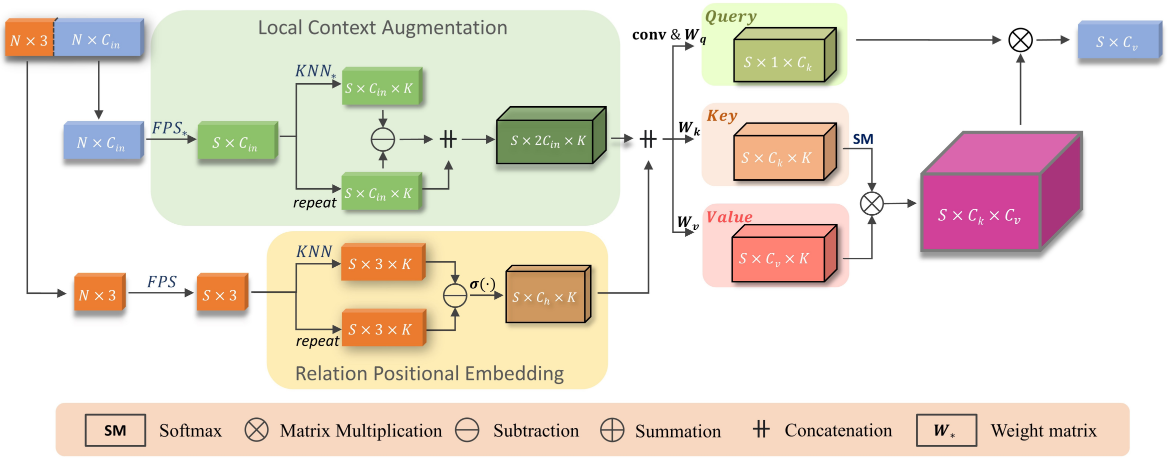


FIGURE 2.5. Detailed architecture of our attention module in 3DMedPT, where KNN and FPS denote k-nearest neighbor and farthest point sampling operations [166], respectively.

2.3.1 Local Context Augmentation

Although self-attention is able to model long-range dependencies over the global domain, it cannot aggregate local information, which is essential in point cloud analysis [66]. However, different from regular layouts such as 2D images where spatially neighboring pixels usually have high semantic correlations, 3D point clouds are unordered and nearby points can have no geometric or semantic relations due to permutation variance. Hence, instead of using local attention [170] which may constrain the model's receptive field, we reform the input feature before each attention layer by defining a local context region, with the assumption that spatially closed points in Euclidean coordinates can have some relations for geometric study. We thus follow the idea of PointNet++ [166] by firstly downsampling the points using farthest point sampling (FPS) and then group features from local contexts as DGCNN [223].

Specifically, given xyz coordinates of N input points $\mathbf{P} = \{\mathbf{p}_i \in \mathbb{R}^3\}_{i=1}^N$ and the corresponding features $\mathbf{F} = \{\mathbf{f}_i \in \mathbb{R}^{C_{in}}\}_{i=1}^N$ with C_{in} feature channels, the whole process can be formulated as follows:

$$\begin{aligned} \mathcal{N}(\mathbf{p}_i) &= \text{KNN}(\mathbf{P}, ||\mathbf{p}_i - \mathbf{p}_j||_2^2),\ \mathbf{p}_j \in \mathbf{P}_S, \\ \mathbf{f}'_i &= [\mathbf{f}_j - \mathbf{f}_i, \mathbf{f}_i]_{j \in \mathcal{N}(\mathbf{p}_i)} \in \mathbb{R}^{K \times 2C_{in}}, \end{aligned} \tag{2.6}$$

where $\mathbf{P}_S$ is a point set downsampled from $\mathbf{P}$ using FPS, KNN$(\cdot)$ is K-nearest neighbor function, $[\cdot, \cdot]$ is concatenation, and $\mathbf{f}'_i$ is the augmented feature with local contexts. In this case, $\mathbf{f}'_i$ can now retain the locality property.

2.3.2 Convolution at Query

This convolution operation proposed at query has two purposes: (1) it allows us to aggregate local responses and update geometric features at query; (2) Since global interaction should be considered between the query and content lambda softmax$(\mathbf{K})^\top\mathbf{V}$, so it is natural to capture global features from query. Therefore, we introduce convolution (i.e., EdgeConv) to Transformer to generate global query information.

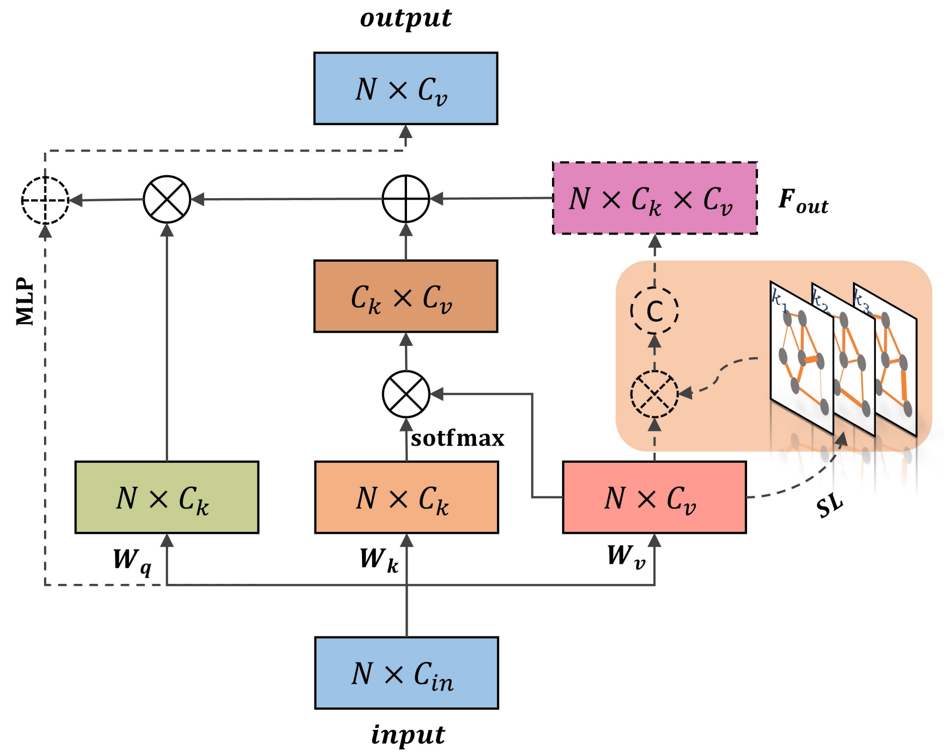


FIGURE 2.6. Modified attention module with MGR, where SL denotes the self-loop and residual connection is applied to compensate the low-level information loss. All dotted components imply different designs with the original Lambda attention [15].

Formally, the query tensor is obtained from the updated features $\mathbf{F}' = \{\mathbf{f}'_i\}_{i=1}^{N} \in \mathbb{R}^{N \times K \times 2C_{in}}$ by using EdgeCov:

$$\mathbf{Q} = \text{EdgeConv}(\mathbf{F}')W_q \in \mathbb{R}^{N \times C_k}, \tag{2.7}$$

where $W_q \in \mathbb{R}^{2C_{in} \times C_k}$. In this case, the local information is integrated into $\mathbf{Q}$, then we apply linear projections on flattened $\mathbf{F}'$ to compute $\mathbf{K}$ and $\mathbf{V}$ as follows:

$$\begin{aligned} \mathbf{K} &= \text{Flatten}(\mathbf{F}')W_k \in \mathbb{R}^{(N \times K) \times C_k}, \\ \mathbf{V} &= \text{Flatten}(\mathbf{F}')W_v \in \mathbb{R}^{(N \times K) \times C_v}, \end{aligned} \tag{2.8}$$

where $W_k \in \mathbb{R}^{2C_{in} \times C_k}$ and $W_v \in \mathbb{R}^{2C_{in} \times C_v}$. Based on the above modifications, we show an improved version of Eq. 2.3 in a per-point form as:

$$\mathbf{y}_i = \mathbf{q}_i \big(\text{softmax}(\mathbf{k}_i)^\top \mathbf{v}_i\big), \tag{2.9}$$

where $\mathbf{y}_i \in \mathbb{R}^{C_v}$ is the layer output, $\mathbf{k}_i \in \mathbb{R}^{K \times C_k}$ and $\mathbf{v}_i \in \mathbb{R}^{K \times C_v}$.

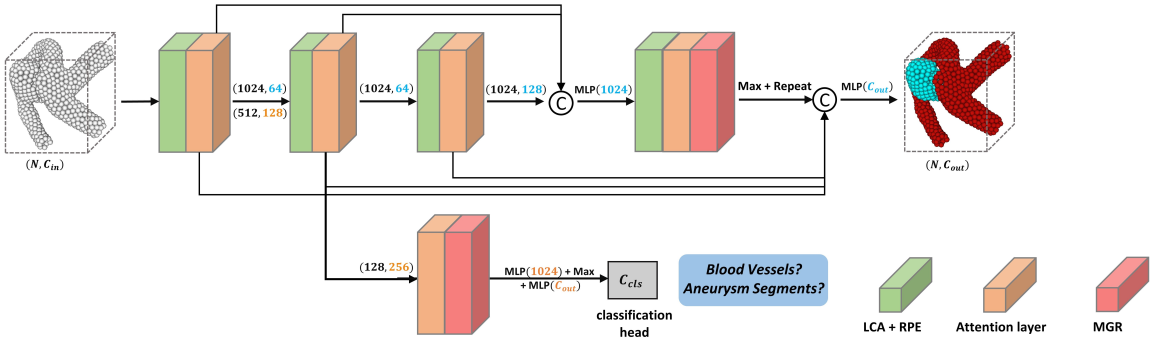


FIGURE 2.7. The overall architecture of 3DMedPT for medical point cloud analysis. Numbers in black, blue, and orange indicate the point number, the feature dimension for classification, and the feature dimension for segmentation. LCA and RPE denote local context augmentation and relative positional embedding, respectively.

2.3.3 Relative Positional Embedding

As mentioned in [266], positional information is critical for 3D point cloud processing. Especially for medical data where the structure is incomplete and complex, embedding positional bias encourages the model to focus on local geometry. In this work, we use relative positional information since computing absolute positions requires storing an ordered list before and after point permutations, which increases the computational overhead. In addition, we integrate the relative positions from local contexts with input feature as we empirically find that using addition cannot give the best result. Therefore, we define learnable relative positional embedding as follows:

$$\begin{aligned} \mathbf{h}_i &= \sigma\big([\mathbf{p}_j - \mathbf{p}_i]_{j \in \mathcal{N}(\mathbf{p}_i)}\big) \in \mathbb{R}^{K \times C_h}, \\ \mathbf{f}'_i &= [\mathbf{f}_j - \mathbf{f}_i, \mathbf{f}_i, \mathbf{h}_i]_{j \in \mathcal{N}(\mathbf{p}_i)}, \end{aligned} \tag{2.10}$$

where $\sigma(\cdot)$ is an MLP.

However, learning accurate positional bias is quite hard when dataset is too small [206], which is our case when dealing with medical data. we therefore apply different MLPs $\sigma(\cdot)$ to learn positional information at query, key, and value positions for accurate and complex medical geometry learning. As illustrated in Figure 2.5, we improve Lambda attention by introducing positional bias terms at a modest cost to address the irregular geometric traits lurking inside medical data.

2.3.4 Multi-Graph Reasoning

Feature representations are critical for model performance, especially when dealing with insufficient training samples, which is often the case in medical domain. In our method, we propose to enrich the feature representations by exploiting global relations of content in deep layers of our attention block with a Multi-Graph Reasoning (MGR) module based on graph reasoning [35] and graph convolutions [104]. As suggested in [140], graphs can be constructed on feature channels by learning graph nodes from channels, which could save the computational cost for the case of large input numbers. In contrast to [140] that only a single graph is established over channels, we design an MGR module to initialize multiple graphs simultaneously with learnable adjacency matrices, enhancing the diversity of node features via various graph states. As shown inside the pink box of Figure 2.6, MGR is adapted on *values* to replace the original positional encoding part in the last attention layer, where neighboring information augmentation is intentionally ignored for global information aggregation and relational interactions. Hence, the output $\mathbf{F}_{out}$ of MGR module can be formulated as:

$$\mathbf{F}_{out} = [\mathrm{ReLU}\left((\mathbf{V}+\mathbf{I})\mathbf{a}_i\right)]_{i\in C_k}, \quad \mathbf{a}_i \in \mathbb{R}^{C_v \times C_v}, \tag{2.11}$$

where $\mathbf{I}$ is the identity matrix, indicating that the self-loop of each node is introduced. C_k graphs and the corresponding adjacency matrices $\mathbf{a}_i$ are concurrently computed from C_v channel nodes, such that contextual interactions between nodes can be modeled and structural relations are captured for feature learning.

2.4 Experiments and Results

In this section, we present our experimental results on medical 3D points (IntrA [249]). We also test the generalization ability on non-medical 3D point clouds (ModelNet40 [233] and ShapeNetPart [252]). Extensive experiments for ablation studies are conducted as well.

2.4.1 Datasets

IntrA. IntrA [249] is a 3D medical point cloud dataset for binary classification and part segmentation to distinguish blood vessels and aneurysms, which contains mesh and point representations of the data structure. It contains 103 3D models of entire brain vessels, which are reconstructed from 2D MRA images of patients, and 1909 blood vessel segments which contains 1694 healthy vessel segments and 215 aneurysm segments for diagnosis. 116 aneurysm segments are divided and annotated manually by medical experts; We use the point cloud with overall 2025 samples for classification. Five-fold cross-validation was adopted with F1-score and per-class testing accuracy as evaluation metrics.

ModelNet40. ModelNet40 [233] has been a benchmark dataset in many deep learning-based 3D data analysis tasks. The dataset comprises 3D CAD models from various categories such as tables, chairs, sofas, beds, and many other common objects, which consists of 13,211 3D synthetic models for general objects, with 9843 training samples and 2468 testing samples ranged within 40 classes. We uniformly sample 1024 points only with 3D coordinates as input features, and shuffle the points as in [166].

ShapeNetPart. ShapeNetPart [252] is a commonly-used benchmark for part segmentation of 3D shapes. The dataset contains 3D models across various categories, and each model is annotated with distinct part labels. For instance, a chair might be segmented into parts such as the backrest, seat, and legs. ShapeNetPart covers 16 shape categories: airplane, bag, cap, car, chair, earphone, guitar, knife, lamp, laptop, motorbike, mug, pistol, rocket, skateboard, and table. In total, it contains 16,880 3D samples with 14,006 training and 2874 testing data, where there are 50 parts across the 16 categories, each category having 2-6 parts. We sample 2048 points from each object and use mean intersection over union averaged across 16 classes (cls. mIoU) as the evaluation metric.

2.4.2 Evaluation Metrics

To evaluate the classification performance of our methods in an unbiased manner, we follow the evaluation metrics proposed in [249] by using F1-score and per-class testing accuracy.

The mathematical expression for the F1 score is:

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{2.12}$$

where Precision is the number of true positive predictions (TP) divided by the sum of the true positive and false positive predictions (FP). Recall is the number of true positive predictions divided by the sum of the true positive and false negative predictions (FN).

The mathematical expression for the per-class testing accuracy for a particular class i is:

$$\text{Accuracy}_i = \frac{\text{TP}_i + \text{TN}_i}{\text{TP}_i + \text{FP}_i + \text{FN}_i + \text{TN}_i}, \tag{2.13}$$

where TN denotes the number of instances of other classes correctly not classified as class i.

For evaluation of segmentation performance, we leverage mean intersection over union (mIoU), which provides an average measure of the overlap between the predicted segmentation and the ground-truth labels for all classes. The mathematical expression for mIoU is:

$$\begin{aligned} \text{mIoU} &= \frac{1}{C} \sum_{c=1}^{C} \text{IoU}(c), \\ \text{IoU}(c) &= \frac{\text{Intersection}(c)}{\text{Union}(c)}, \end{aligned} \tag{2.14}$$

where Intersection(c) is the number of pixels/voxels that are correctly predicted for class c, and IoU(c) is the total number of pixels/voxels that are actually of class c plus the number of pixels/voxels that are wrongly predicted as class c. C is the number of total classes.

2.4.3 Implementation Details

The overall model architecture for 3D object classification and part segmentation is shown in Figure 2.7. In our overall network model, the input feature dimension C_{in} is set to 3 representing 3D normal vectors, and if there is no normal vectors, C_{in} will be the xyz positions. The output feature dimension C_{out} is set to 2 for classifying and segmenting blood vessels or aneurysms, with binary cross entropy as the loss function. As shown in Figure 2.7, we follow the architecture of PointNet++ for classification, where points are downsampled and features are embedded with locality, followed by the attention module to enhance interactions between local contexts and global contents. MGR is applied to further enrich the feature representations with graph reasoning, and maxpooling is employed to summarize the information while being insensitive to permutation as a symmetric function. For segmentation, we adopt a DGCNN-like network. Each MLP block contains a linear mapping layer, batch normalization layer and ReLU. To address bottleneck issues caused by a small C_v, the multi-query algorithm [15] is utilized so that h different query heads are constructed and concatenated, deriving a new output feature $\mathbf{y}_i \in \mathbb{R}^{hC_v}$.

2.4.4 3D Object Classification

We first examine the model behavior on IntrA and then investigate our model on ModelNet40.

IntrA. As shown in Table 2.1, our method reaches the highest accuracies of 94.06% with 512 points for aneurysm (A.) and 99.24% with 1024 points for blood vessel (V.) detection by using the PointNet++ backbone, which overpass the original work by 0.7% and 6.3% respectively. Our 3DMedPT achieves the best F1 score of 0.936 with 1024 points, which outperforms its Transformer counterpart PCT by 2.4%. It is also 3.3% and 9.1% higher than the latest attention-based works: PAConv [242] and AdaptConv [267], showing the superiority of our method on 3D medical point cloud analysis.

ModelNet40. Our performance compared to other SoTA methods is listed in Table 2.2. It can be observed that we achieve an accuracy of 93.4%, which overpasses most typical point-based designs [241, 231, 223]. Additionally, our model performance is better than some networks based on attention algorithms (e.g., Set Transformer [111], PAT [248] and Point2Sequence [126]), and our model is also better than the Transformer counterpart PCT [66] with a much smaller model size (see Table 2.6). Our classification accuracy is lower than recently proposed Point Transformer [266] by only 0.3%, while this small gap validates the good generalization ability of 3DMedPT. Hence, our design can not only deal with medical dataset with complex topology such as blood vessels or aneurysms, but also distribute the excellence to regular 3D shapes.

TABLE 2.1. Classification results of per-class accuracy and F1-score on healthy vessel segments (V.) and aneurysm segments (A.) with all input features. Results are averaged across all 5 folds.

Method	#Points	V. (%)	A. (%)	F1
PointNet [164]	512	94.45	67.66	0.691
	1024	94.98	64.96	0.684
	2048	93.74	69.50	0.692
PointNet++ [166]	512	98.52	86.69	0.893
	1024	98.52	88.51	0.903
	2048	98.76	87.31	0.902
PointCNN [118]	512	98.38	78.25	0.849
	1024	98.79	81.28	0.875
	2048	98.95	85.81	0.904
PointConv [231]	512	99.21	91.96	0.915
	1024	98.89	83.57	0.883
	2048	98.61	90.47	0.883
SO-Net [115]	512	98.76	84.24	0.884
	1024	98.88	81.21	0.868
	2048	98.88	83.94	0.885
SpiderCNN [245]	512	98.05	84.58	0.869
	1024	97.28	87.90	0.872
	2048	97.28	84.89	0.866
DGCNN [223]	512	95.22	60.73	0.658
	1024	95.34	72.21	0.738
	2048	97.93	83.40	0.859
GS-Net [241]	512	98.55	83.84	0.873
	1024	98.78	83.08	0.872
	2048	98.39	85.74	0.882
PCT [66]	512	99.03	89.07	0.911
	1024	98.87	89.71	0.914
	2048	98.96	89.49	0.917
PAConv [242]	512	98.53	89.00	0.904
	1024	98.98	89.71	0.906
	2048	98.19	85.74	0.882
AdaptConv [267]	512	97.58	79.99	0.809
	1024	99.05	82.90	0.858
	2048	97.87	75.94	0.799
3DMedPT	512	99.02	**94.06**	0.920
	1024	**99.24**	93.26	**0.936**
	2048	99.07	93.49	0.931

TABLE 2.2. Classification results on ModelNet40 with different input types and point numbers.

Method	Input	#Points	Acc. (%)
Set Transformer [111]	xyz	5k	90.4
PointCNN [118]	xyz	1k	91.7
DGCNN [223]	xyz	1k	92.2
Point2Sequence [126]	xyz	1k	92.6
GS-Net [241]	xyz	1k	92.9
RS-CNN [129]	xyz	1k	92.9
SO-Net [115]	xyz	2k	90.9
KPConv [199]	xyz	7k	92.9
PCT [66]	xyz	1k	93.2
AdaptConv [267]	xyz	1k	93.4
PAConv [242]	xyz	1k	93.6
Point Transformer [266]	xyz	1k	**93.7**
PAT [248]	xyz + norm	1k	91.7
PointConv [231]	xyz + norm	1k	92.5
PointASNL [247]	xyz + norm	1k	93.2
PointNet++ [166]	xyz + norm	5k	91.9
SpiderCNN [245]	xyz + norm	5k	92.4
3DMedPT	xyz	1k	93.4

2.4.5 3D Part Segmentation

We then validate the segmentation ability of our model on both IntrA and ShapeNetPart, with the same data augmentation method as Secection 2.4.4.

IntrA. There are a total of 116 annotated samples for part segmentation task in IntrA, where the boundary lines are grouped into aneurysm segments, making it a binary segmentation task. Five-fold cross-valuation is still applied with evaluation metrics based on Point Intersection over Union (IoU) and Sørensen–Dice cefficient (DSC). Results are reported in Table 2.3. It can be seen that we achieve the highest IoU and DSC values of 94.82% and 97.29% for parent vessels segmentation with 512 input points. Meanwhile, our 3DMedPT also has the best performance on the aneurysm segmentation with 1024 points, resulting in IoU and DSC values of 82.39% and 89.71%. Our work outperforms PAConv and AdaptConv by a large margin by 4.7% and 9.5% on A. IoU and 2.8% and 4.2% on V. IoU, which exhibits our superiority on medical data.

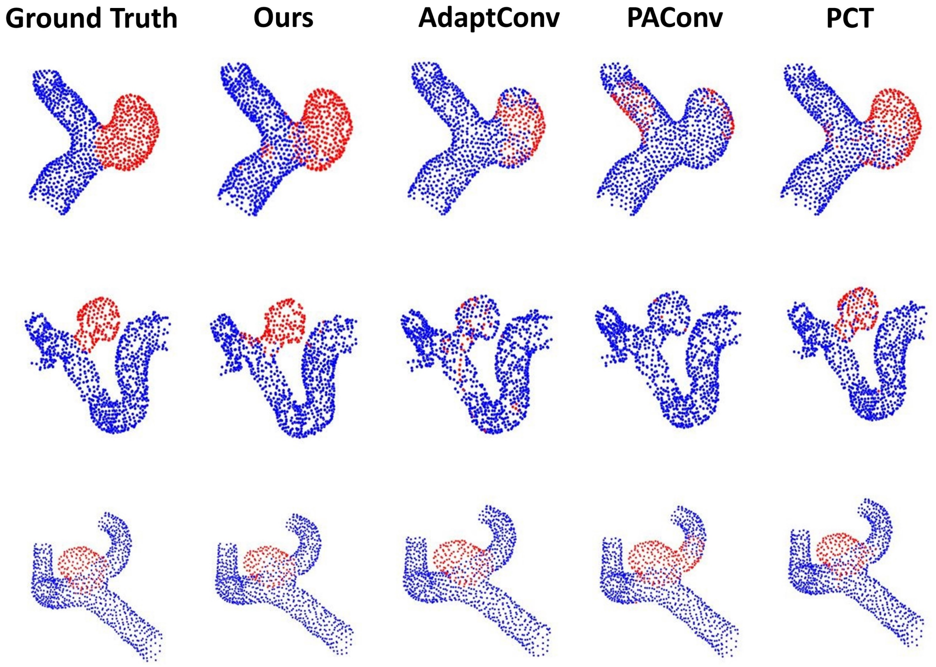


FIGURE 2.8. Qualitative comparisons on IntrA segmentation. Ground-truth samples are shown in the first column for reference.

To further examine the model behavior, we qualitatively evaluate our approach with respect to some recent works such as PAConv. Ground-truth samples are shown in the 1st column for reference in Figure 2.8. As can be seen, when the aneurysm takes a large size ratio of the blood vessel (row 1), our model performs the best and PCT cannot fully understand the shape of aneurysm, while PCT gives the similar segmentation results as ours in other cases (rows 2-3). However, we can see that the latest work PAConv totally fails when complicated structures are encountered (row 2).

More visual results on the segmentation of IntrA are shown in Figure 2.9 based on our best model, which are compared to the corresponding ground-truth annotations for comprehensiveness. As shown in Figure 2.9, our method achieves fairly precise segmentation results on most cases, however, few undesired results might appear especially when the size ratio of aneurysms becomes smaller than the healthy blood vessels, or when the 3D structure topology becomes complicated.

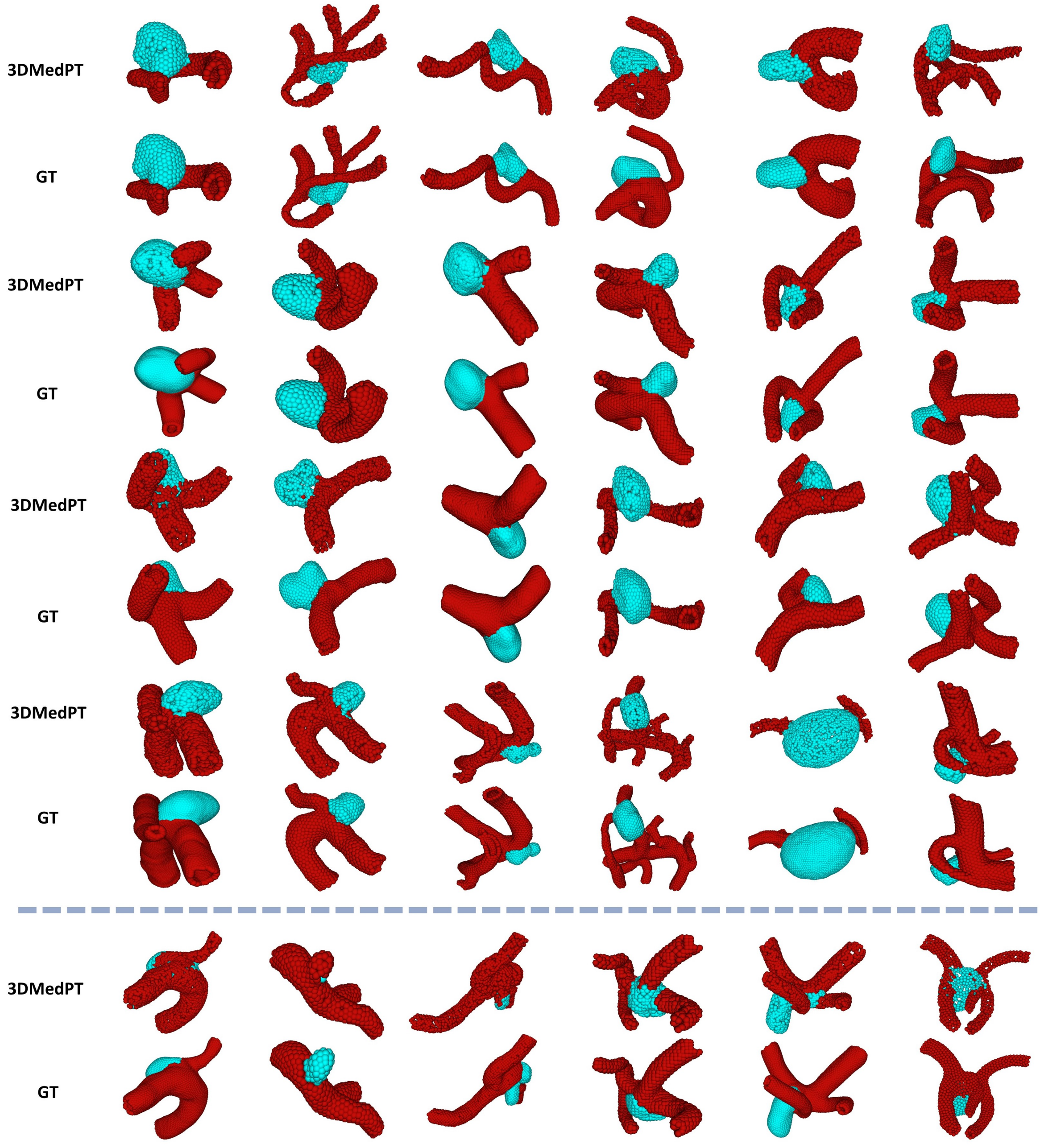


FIGURE 2.9. Segmentation comparisons between ground-truth annotations and the outputs generated from the 3DMedPT on IntrA dataset. Good segmentation results are shown above the red line and some failed cases are shown below the red line.

ShapeNetPart. Table 2.4 presents detailed per-class results and the overall cls. mIoU of our DGCNN backbone. We achieve the overall value of 84.3%, which is 0.4% lower than PAConv but 1.1% higher than AdaptConv. Although we cannot achieve the best result, our model

TABLE 2.3. Segmentation results of each point-based network. V. and A. represent parent vessel segments and aneurysm segments.

Method	#Points	IoU (%)		DSC (%)	
		V.	A.	V.	A.
PointNet [164]	512	73.99	37.30	84.05	48.96
	1024	75.23	37.07	85.00	48.38
	2048	74.22	37.75	84.17	49.59
PointNet++ [166]	512	93.42	76.22	96.48	83.92
	1024	93.35	76.38	96.47	84.62
	2048	93.24	76.21	96.40	84.64
PointCNN [118]	512	92.49	70.65	95.97	78.55
	1024	93.47	74.11	96.53	81.74
	2048	93.59	73.58	96.62	81.36
SO-Net [115]	512	94.22	80.14	96.95	87.90
	1024	94.42	80.99	97.06	88.41
	2048	94.46	81.40	97.09	88.76
SpiderCNN [245]	512	90.16	67.25	94.53	75.82
	1024	87.95	61.60	93.24	71.08
	2048	87.02	58.32	92.17	67.74
PointConv [231]	512	94.16	79.09	96.89	86.01
	1024	94.59	79.42	97.15	86.29
	2048	94.65	79.53	97.18	86.52
GS-Net [241]	512	90.06	64.48	94.62	74.54
	1024	90.93	66.29	95.10	78.85
	2048	91.06	65.76	95.15	75.06
PCT [66]	512	92.49	78.09	96.08	85.84
	1024	92.05	78.12	95.85	86.77
	2048	91.66	77.10	95.43	86.02
AdaptConv [267]	512	90.45	70.25	96.01	80.60
	1024	90.69	75.26	94.92	84.40
	2048	90.97	75.08	95.05	84.72
PAConv [242]	512	91.97	78.66	95.66	87.57
	1024	90.34	74.31	94.54	83.16
	2048	92.20	70.59	95.81	79.18
3DMedPT	512	**94.82**	81.80	**97.29**	89.25
	1024	94.76	**82.39**	97.25	**89.71**
	2048	93.52	80.13	96.59	88.69

shows great performance in the class-wise segmentation results where we achieve the best in cap and mug. Considering the performance gaps among ours, PAConv and AdaptConv on the IntrA segmentation task, we claim that our model can generalize well to general datasets. Qualitative results are shown in Figure 2.10.

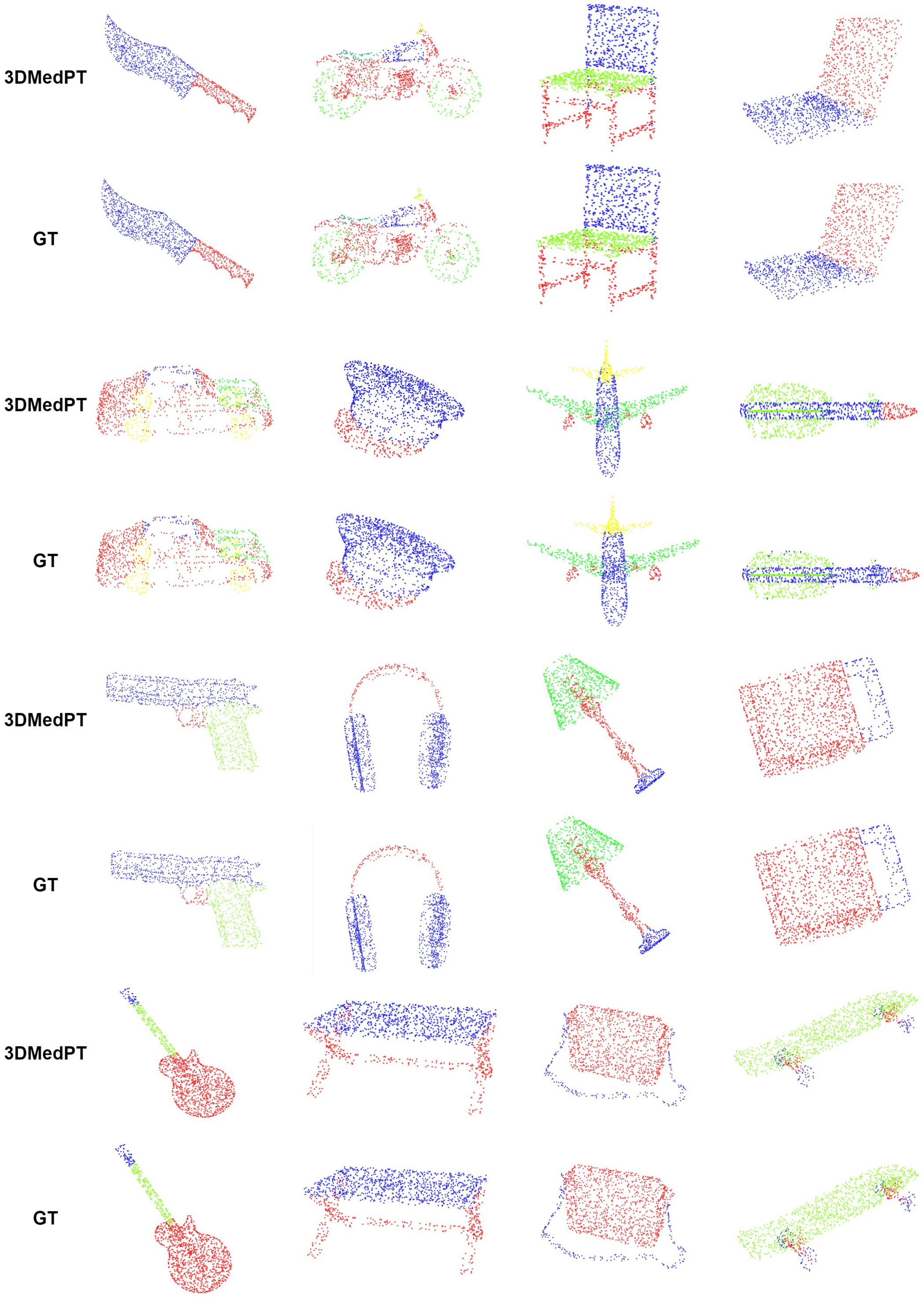


FIGURE 2.10. Segmentation comparisons between ground-truth annotations and the outputs generated from the 3DMedPT on ShapeNetPart dataset.

TABLE 2.4. Segmentation results of different methods on ShapeNetPart.

Method	cls. mIoU	air plane	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor bike	mug	pistol	rocket	skate board	table
PointNet [164]	80.4	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
PointNet++ [166]	81.9	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
PointCNN [118]	84.6	84.1	**86.5**	86.0	80.8	90.6	79.7	**92.3**	88.4	85.3	96.1	77.2	95.2	84.2	64.2	80.0	83.0
DGCNN [223]	82.3	84.0	83.4	86.7	77.8	90.6	74.7	91.2	87.5	82.8	95.7	70.8	94.6	81.1	63.5	74.5	82.6
KPConv [199]	**85.0**	83.8	86.1	88.2	**81.6**	91.0	80.1	92.1	87.8	82.2	96.2	**77.9**	95.7	**86.8**	**65.3**	**81.7**	83.6
PointASNL [247]	83.4	84.1	84.7	87.9	79.7	92.2	73.7	91.0	87.2	84.2	95.8	74.4	95.2	81.0	63.0	76.3	83.2
RS-CNN [129]	84.0	83.5	84.8	88.8	79.6	91.2	81.1	91.6	88.4	86.0	96.0	73.7	94.1	83.4	60.5	77.7	83.6
PCT [66]	83.1	**85.0**	82.4	89.0	81.2	**91.9**	71.5	91.3	88.1	**86.3**	95.8	64.6	95.8	83.6	62.2	77.6	83.7
PAConv [242]	84.6	84.3	85.0	90.4	79.7	90.6	80.8	92.0	**88.7**	82.2	95.9	73.9	94.7	84.7	65.9	81.4	84.0
AdaptConv [267]	83.4	84.8	81.2	85.7	79.7	91.2	80.9	91.9	88.6	84.8	**96.2**	70.7	94.9	82.3	61.0	75.9	**84.2**
3DMedPT	84.3	81.2	86.0	**91.7**	79.6	90.1	**81.2**	91.9	88.5	84.8	96.0	72.3	**95.8**	83.2	64.6	78.2	83.8

TABLE 2.5. Positional embeddings in 3DMedPT evaluated on F1-score. RPE_k, RPE_q and RPE_v indicate relative positional embeddings at the key, query and value.

	RPE_q	RPE_k	RPE_v	F1
A				0.905
B	✓			0.924
C		✓		0.915
D			✓	0.921
E	✓		✓	0.920
F		✓	✓	0.917
G	✓	✓		0.928
H	✓	✓	✓	0.936

2.4.6 Ablation Study

In this section, we first examined the contribution of positional embeddings, and then investigated our model robustness on noise and compared the efficiency with some typical methods from Table 2.1. Unless specified, all experiments are conducted on IntrA dataset for the 3D object classification with the first fold as the testing set and the others as the training set, and 1024 points are sampled for fast computation.

Positional Embeddings. To investigate the effectiveness of the relative positional embedding, different models are established and F1 scores are averaged across all folds and reported in Table 2.5. Model A is the design where no positional bias is introduced in the attention block, and we examine the cases when positional embedding is shared at all three positions individually (models B $\rightarrow$ D). More investigations are done when positional embedding is introduced at any two positions (model E $\rightarrow$ G) or query, key, and value (model H).

We can see from Table 2.5 that without embedding position information, our model can still achieve a reasonable performance due to the design of our local context augmentation module. Besides, introducing positional bias does improve the model performance when comparing model A with any others. We find that introducing positional bias terms to all positions (model H) gives us the best result with the F1 score of 0.936 across all 5 folds, which indicates that more accurate positional information is learned via learning mapping in all query, key and value positions.

TABLE 2.6. Model complexity of 3DMedPT for the IntrA classification, where model parameters are in the unit of millions and throughput is reported in examples per second.

Method	#Params	Throughput	F1
PointNet++ [166]	1.75M	245 ex/s	0.769
DGCNN [223]	1.81M	365 ex/s	0.738
PCT [66]	2.73M	471 ex/s	0.872
AdaptConv [267]	1.76M	230 ex/s	0.806
PAConv [242]	2.32M	507 ex/s	0.866
3DMedPT	**1.54**M	**843** ex/s	**0.922**

Model Efficiency. Computational costs of different models compared with 3DMedPT were explored in terms of the model size and processing speed. As shown in Table 2.6, we achieved the highest performance with the smallest model size which only contains 1.54M model parameters, processing 843 examples per second. Although the processing speed of PointNet [164] is the fastest, it cannot perform as well as other models at a lower computational speed. Moreover, our Transformer counterpart PCT [66] requires more trainable parameters to achieve a relatively good performance, with a slower processing speed than ours.

Robustness Analysis. We demonstrated our model robustness to the point density by using sparser points as the network input from 2048 to 128 points. We compared our results with several works in Figure 2.11 (left), where all networks were trained with 1k points on IntrA. The absolute dropping difference from 2048 to 128 points of our method is 6.7%, which is the same as our Transformer counterpart PCT, while we reached the best F1 score on all experiments with different numbers of input points. For the noise resistance investigation, we introduced different numbers of noisy 3D points with random positions during model testing following [247]. As can be seen from Figure 2.11 (right), 3DMedPT is more robust to noise compared with some latest works PAConv [242] and AdaptConv [267] under all testing environments. The absolute difference between no noise and 50 noisy points for our model is 11.3%, which is smaller than PCT with the value of 14.5%, presenting our excellent robustness ability to the noise.

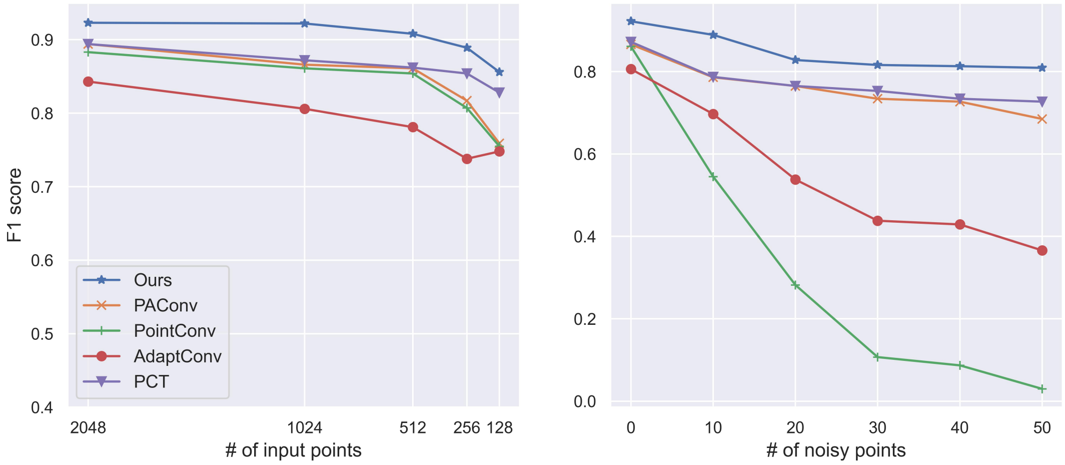


FIGURE 2.11. **Left**: Comparison on different numbers of input points. **Right**: Comparison on different numbers of noisy points.

2.5 Discussion

We design the model in a way that only deals with 3D structures, and our design only focuses on the difference between 3D medical models and general models (i.e., cars, airplanes). This is the limitation when applying our model to different medical data.

In addition, we have conducted different experiments on 3D point clouds, we follow PointNet [164] to implement a sanity check on the proposed method with experiments on a 2D medical data RetinalOCT [100].

RetinalOCT involves 2D gray-scale images of retinal diseases, which is comprised of 4 diagnosis categories with 108,318 training and 1,000 testing samples from 633 patients, leading to a multi-class classification task. To make it compatible for training with our model, we convert 2D image pixels to 2D points. Specifically, we firstly resize each image to 256x256 and use Sobel filters to detect the edges of each image, with (x, y) assigned by (row, col) of each pixel and $z = 0$. The number of points for each data sample is fixed and determined by the average value of all point sets obtained from all images. During training, we normalize each point set to a unit cube of [-1, 1] and use data augmentation following IntrA.

In Table 2.7, it can be seen that when compared with ResNet-18/50 baselines [75] with 3D convolutions and an open-source AutoML tool (AutoKeras [92]). The highest testing

TABLE 2.7. Classification results on RetinalOCT with different methods.

Method	Input Type	Acc. (%)
Pre-trained InceptionV3 [100]	pixels	96.6
ResNet-18 [75]	pixels	95.8
ResNet-50 [75]	pixels	96.1
AutoKeras [92]	pixels	96.3
PointNet [164]	points	87.5
DGCNN [223]	points	87.9
PCT [66]	points	87.2
3DMedPT	points	87.9

accuracy is achieved by [100] where InceptionV3 [196] is applied and pre-trained on ImageNet. Although we can outperform PointNet, DGCNN, and PCT in medical point clouds dataset, we can only achieve similar performance with these methods on RetinalOCT. Moreover, there are noticeable performance gaps between our approach and CNN-based methods. We argue that it is the information loss due to the data type conversion (i.e., image resizing and ignorance of pixel intensity) that causes the performance difference. In the future, we plan to dvelop a universal method that can work well on both 2D and 3D medical datasets.

2.6 Summary

In this chapter, we propose a Transformer network for 3D medical point cloud analysis, namely 3DMedPT, which can model long-range dependencies of global contents via the convolutional operation introduced at query to summarize local feature responses, and local context interactions based on lambda attention modified with local context augmentation. Variant relative positional information for query, key and value is encoded to capture the complex structure of medical data. Global interactions between features are obtained from channel space where multiple graphs are constructed to model diverse graph states, improving the expressiveness of feature information. Our model performs the best in 3D medical object classification and part segmentation tasks. Moreover, extensive analyses on general 3D point cloud datasets have validated the good generalization ability of our model.

From a larger perspective, the segmentation and classification tasks for the medical point clouds are not limited to intracranial aneurysm treatment. With the significant improvement proposed by our method, we believe that it will contribute to the medical or biomedical domains.

CHAPTER 3

Deep Learning-based Analysis for Rotated Point Clouds

Besides dealing with medical point clouds, we also examine that using existing 3D deep learning-based method, it is hard for them to classify the raw point clouds that are not in the canonical poses. As mentioned in Section 1.1, the pose information of scanned point clouds is normally unknown in real-world scenarios. Existing deep learning-based methods [164, 166] are unable to process randomly rotated point clouds, as they are trained on 3D objects of canonical poses. Hence, methods that are able to extract features invariant to random rotations are in urgent need. Recent investigations on rotation invariance for 3D point clouds have been devoted to devising rotation invariant feature descriptors or learning canonical spaces where objects are semantically aligned. Examinations of learning frameworks for invariance have seldom been looked into.

In this chapter, we propose a novel method to extract rotation invariant shape features given randomly rotated point clouds. Specifically, we review the rotation invariance in terms of point cloud registration and propose an effective framework for rotation invariance learning via three sequential stages, namely rotation invariant shape encoding, aligned feature integration, and deep feature registration. We first encode shape descriptors constructed with respect to reference frames defined over different scales, e.g., local patches and global topology, to generate rotation invariant latent shape codes. Within the integration stage, we propose Aligned Integration Transformer to produce a discriminative feature representation by integrating point-wise self- and cross-relations established within the shape codes. Meanwhile, we adopt rigid transformations between reference frames to align the shape codes for feature consistency across different scales. The features are integrated in a deep level and registered to both rotation invariant shape codes to maximize feature similarities, so that rotation invariance is preserved and shared semantic information is implicitly extracted from shape codes. Experimental results on rotated point cloud datasets show that our proposed methods outperform SoTA models by a large margin.

3.1 Introduction

With the development of recent deep learning models for 3D point clouds process, it is still difficult to directly apply these models to real world data because raw 3D point clouds are normally captured from different viewing angles, resulting in unaligned data samples, which inevitably impact the deep learning models which are sensitive to rotations. Therefore, rotation invariance becomes an important research topic in the 3D domain.

To address the problem, a straightforward way is to directly learn a model given raw 3D objects to augment training data with massive rotations, which however requires a large memory capacity and exhibits limited generalization ability to unseen data of random poses [103]. There are attempts to align a 3D object to a canonical pose [89, 43], or to learn rotation robust features via equivariance [49, 138], while these methods are not rigorously rotation invariant and present noncompetitive performance on 3D shape analysis. To maintain consistent model behavior under random rotations, some methods [263, 28, 239] follow [56] to handcraft rotation invariant point-pair features. Others [262, 114, 264] design robust features from equivariant orthonormal bases.

In this section, we propose our framework in Figure 3.1 with three sequential stages, namely rotation-invariant shape encoding, aligned feature integration, and deep feature registration. Firstly, we **(a)** construct and feed point pairs with different scales as model inputs, where we consider local patches $\mathbf{P}^{\ell}$ with small number of points and global shape $\mathbf{P}^{g}$ with the whole 3D points. Hence, the final feature representation can be enriched by information from different scales. Low-level rotation-invariant descriptors are thus built on reference frames and encoded to generate latent shape codes $\mathbf{F}^{\ell}$ and $\mathbf{F}^{g}$ following recent point cloud registration (PCR) work [160]. Secondly, we **(b)** introduce a variant of transformer [210], Aligned Integration Transformer (AIT), to implicitly integrate information from both self- and cross-attention branches for effective feature integration. In this way, information encoded from different point scales is aggregated to represent the same 3D object. Moreover, we consider $\mathbf{F}^{\ell}$ and $\mathbf{F}^{g}$ as *unaligned* since they are encoded from *unaligned* reference frames. To address the problem, we follow the evaluation technique proposed in PCR, where we use relative rotation information ($\mathbf{T}$) with learnable layers to align $\mathbf{F}^{\ell}$ and $\mathbf{F}^{g}$ for feature consistency. Finally, to ensure rotation invariance of the integrated feature U, we follow PCR to **(c)** examine the correspondence map of ($\mathbf{F}^{g}$, U) and ($\mathbf{F}^{\ell}$, U), such that the mutual information between a local patch of a 3D object and the whole 3D object is maximized, and rotation invariance is further ensured in the final geometric feature.

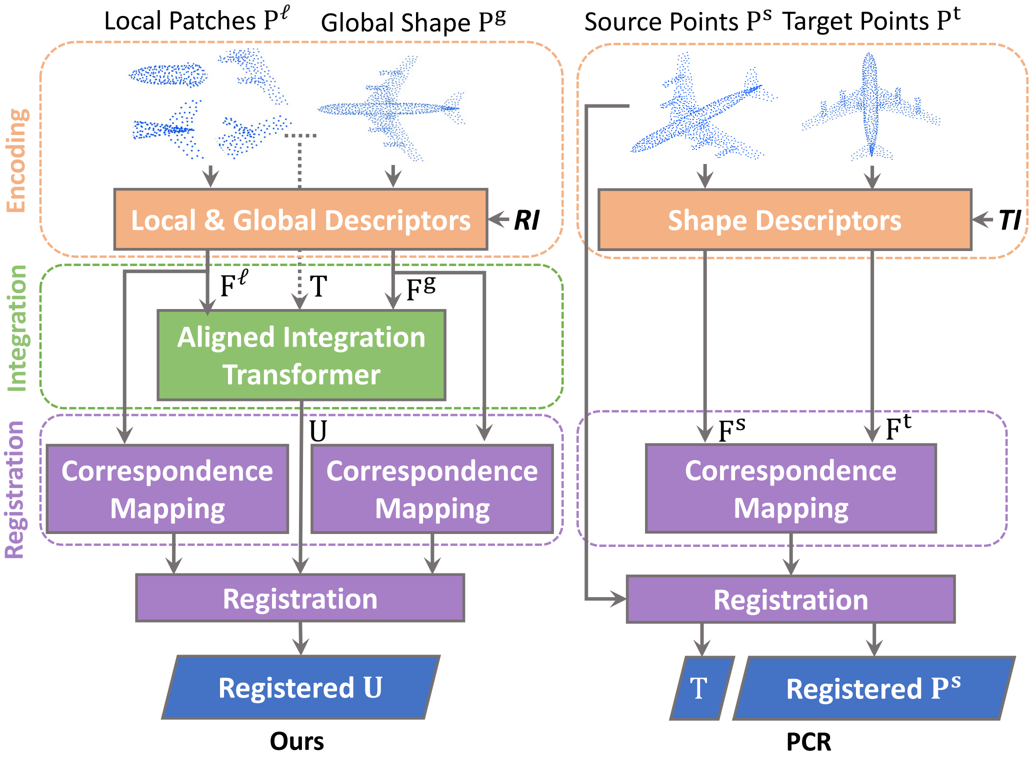


FIGURE 3.1. Frameworks of our design (left) and robust point cloud registration (right), where TI and RI are transformation invariance and rotation invariance, and $\mathbf{T}$ is the rigid transformation. The dotted line indicates the computation of $\mathbf{T}$ between reference frames.

The contributions of our work are summarized as follows:

(1) To our knowledge, we are the first in developing a PCR-cored representation learning framework towards effective rotation invariance studies on 3D point clouds.

(2) We introduce Aligned Integration Transformer (AIT), a transformer-based architecture to conduct aligned feature integration for a comprehensive geometry study from both local and global scales.

(3) We propose a registration loss to maintain rotation invariance and discover semantic knowledge shared in different parts of the input object.

3.2 Literature Review

3.2.1 Rotation Robust Feature Learning

Networks that are robust to rotations can be equivariant to rotations. Works such as [59, 43] project 3D data into a spherical space for rotation equivariance and perform convolutions in terms of spherical harmonic bases. Others [192, 194] learn canonical spaces to unify the pose of point clouds. Recent works [138, 49, 93] vectorize the scalar activations and mapping SO(3) actions to a latent space for easy manipulations. Although these works present competitive results, they cannot be strictly rotation-invariant. Another way for rotation robustness is to learn rotation-invariant features. Handcrafted features can be rotation-invariant [263, 28, 33, 239], but they normally ignore the global overview of 3D objects. Others use rotation-equivariant local reference frames (LRFs) [262, 198, 103] or global reference frames (GRFs) [114] as model inputs based on principal component analysis (PCA). However, they may produce inconsistent features across different reference frames, which would limit the representational power. In contrast to abovementioned methods with rotation robust model inputs or modules, we examine the relation between rotation invariance and PCR and propose an effective framework.

3.2.2 3D Point Cloud Registration

In our thesis, we only consider about rigid registration assuming that 3D objects would not be deformed during the classification or segmentation processes. Specifically, given a pair of LiDAR scans, 3D PCR requires an optimal rigid transformation to best align the two scans.

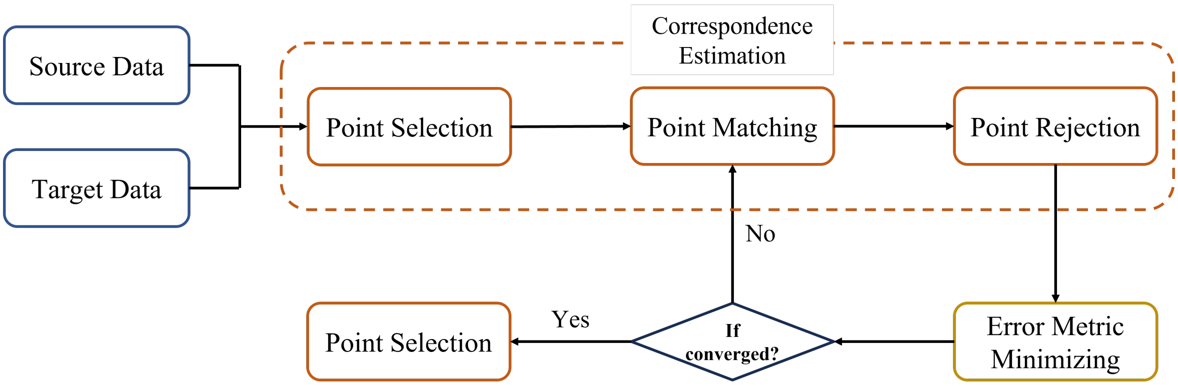


FIGURE 3.2. The pipeline of a general registration method (i.e., iterative closest point (ICP) [16]).

Mathematically, we consider two point clouds: source data $P = \{p_1, p_2, ..., p_n | p \in \mathbb{R}^3\}$ and target data $Q = \{q_1, q_2, ..., q_n | q \in \mathbb{R}^3\}$. The task of rigid registration is to find a transformation to align P with Q, and the transformation consists of a rotation matrix $\mathbf{R}$ and translation matrix $\mathbf{T}$. The mathematical process is shown as follows:

$$\begin{aligned} \text{algorithm: } & \mathbf{R}, \mathbf{T} = \text{Reg\_func}(P, Q) \\ \text{final result: } & P' = \{p'_i\} = \{\mathbf{R} \cdot p_i + \mathbf{T}\},\ i = 1, 2, ..., n \end{aligned} \tag{3.1}$$

where Reg_func is a rigid registration function that estimates the rotation and translation matrices based on P and Q.

Despite the recent emerging of ICP-based methods [16, 222] as shown in Figure 3.2, we follow robust correspondence-based approaches in our work [50, 257, 167, 160], where rotation invariance is widely used to mitigate the impact of geometric transformations during feature learning. Specifically, both [160] and [167] analyze the encoding of transformation-robust information and introduce a rotation-invariant module with contextual information into their registration pipeline. All these methods showing impressive results are closely related to rotation invariance. We hypothesize that the learning framework of rotation invariance can be similar to PCR, and we further prove in experiments that our network is feasible and able to achieve competitive performance on rotated point clouds.

3.2.3 Contrastive Learning with 3D Visual Correspondence

Based on visual correspondence, contrastive learning aims to train an embedding space where positive samples are pushed together whereas negative samples are separated away [74]. The definition of positivity and negativity follows the visual correspondence maps, where pairs with high confidence scores are positive otherwise negative. Visual correspondence is important in 3D tasks, where semantic information extracted from matched point pairs improves the network's understanding on 3D geometric structures. For example, PointContrast [238] explores feature correspondence across multiple views of one 3D point cloud with InfoNCE loss [207], increasing the model performance for downstream tasks. Info3D [183] and CrossPoint [1] minimize the semantic difference of point features under different poses. We follow the same idea by registering the deep features to rotation-invariant features at intermediate levels, increasing feature similarities in the embedding space to ensure rotation invariance.

3.3 Methods

Given a 3D point cloud including N_{in} points with xyz coordinates $\mathbf{P} = \{p_i \in \mathbb{R}^3\}_{i=1}^{N_{in}}$, we aim to learn a shape encoder f that is invariant to 3D rotations: $f(\mathbf{P}) = f(\mathbf{RP})$, where $\mathbf{R} \in SO(3)$ and SO(3) is the rotation group. RI can be investigated and achieved through three stages, namely rotation-invariant shape encoding (Section 3.3.1), aligned feature integration (Section 3.3.2), and deep feature registration (Section 3.3.3).

3.3.1 Rotation-Invariant Shape Encoding

In this part, we first construct the input point pairs from local and global scales based on reference frames, following the idea of [160] to obtain low-level rotation-invariant shape descriptors from LRFs and GRF directly. Then we obtain latent shape codes via two set abstraction layers as in PointNet++ [166].

Rotation Invariance for Local Patches. To construct rotation-invariant features on LRFs, we hope to construct an orthonormal basis for each LRF as $p \in \mathbb{R}^{3\times 3}$. Given a point p_i and its neighbor $p_j \in \mathcal{N}(p_i)$, we choose $\overrightarrow{x_i}^\ell = \overrightarrow{p_m p_i} / \|\overrightarrow{p_m p_i}\|_2$, where p_m is the barycenter of the local geometry and $\| \cdot \|_2$ is L2-norm. We then define $\overrightarrow{z_i}^\ell$ following [201] to have the same direction as an eigenvector, which corresponds to the smallest eigenvalue via eigenvalue decomposition (EVD):

$$
\begin{aligned}
\mathbf{\Sigma}_i^\ell &= \sum_{j=1}^{|\mathcal{N}(p_i)|} \alpha_j \left(\overrightarrow{p_i p_j}\right) \left(\overrightarrow{p_i p_j}\right)^\top , \\
\alpha_j &= \frac{d - \|\overrightarrow{p_i p_j}\|_2}{\sum_{j=1}^{|\mathcal{N}(p_i)|} d - \|\overrightarrow{p_i p_j}\|_2},
\end{aligned}
\tag{3.2}
$$

where α_j is a weight parameter, allowing nearby p_j to have large contribution to the covariance matrix, and d is the maximum distance between p_i and p_j. Finally, we define $\overrightarrow{y_i}^\ell$ as $\overrightarrow{z_i}^\ell \times \overrightarrow{x_i}^\ell$. RI is introduced to p_i with respect to its neighbor p_j as $p_{ij}^\ell = \overrightarrow{p_i p_j}^\top \mathbf{M}_i^\ell$. The latent shape code $\mathbf{F}^\ell \in \mathbb{R}^{N\times C}$ is obtained via PointNet++ and max-pooling.

Rotation Invariance for Global Shape. We apply PCA as a practical tool to obtain rotation invariance in a global scale. Similar to Eq. 3.2, PCA is performed by $\frac{1}{N_0} \sum_{i=1}^{N_0} (\overrightarrow{p_m p_i})(\overrightarrow{p_m p_i})^\top = \mathbf{U}^g \mathbf{\Lambda}^g \mathbf{U}^{g\top}$, where p_m is the barycenter of $\mathbf{P}$, $\mathbf{U}^g = [\overrightarrow{u_1}^g, \overrightarrow{u_2}^g, \overrightarrow{u_3}^g]$ and $\mathbf{\Lambda}^g = \mathrm{diag}(\lambda_1^g, \lambda_2^g, \lambda_3^g)$ are eigenvector and eigenvalue matrices. We take $\mathbf{U}^g$ as the orthonormal basis $\mathbf{M}^g = [\overrightarrow{x}^g, \overrightarrow{y}^g, \overrightarrow{z}^g]$ for GRF. By transforming point p_i with $\mathbf{U}^g$, the shape pose is canonicalized as $p_i^g = p_i \mathbf{M}^g$. Proof of the rotation invariance of p_i^g is omitted for its simplicity, and $\mathbf{F}^g \in \mathbb{R}^{N\times C}$ is obtained following PointNet++.

Sign Ambiguity. EVD introduces sign ambiguity for eigenvectors, which negatively impacts the model performance [22]. The description of sign ambiguity states that for a random eigenvector $\overrightarrow{u}$, $\overrightarrow{u}$ and $\overrightarrow{u}'$, with $\overrightarrow{u}'$ having an opposite direction to $\overrightarrow{u}$, are both acceptable solutions to EVD. To tackle this issue, we simply force $\overrightarrow{z_i}^\ell$ of LRF to follow the direction of $\overrightarrow{op_i}$, with o being the origin of the world coordinate. We disambiguate basis vectors in $\mathbf{M}^g$ by computing an inner product with $\overrightarrow{p_m p_i}, \forall i \in N_0$. Taking $\overrightarrow{x}^g$ for example, its direction is conditioned on the following term:

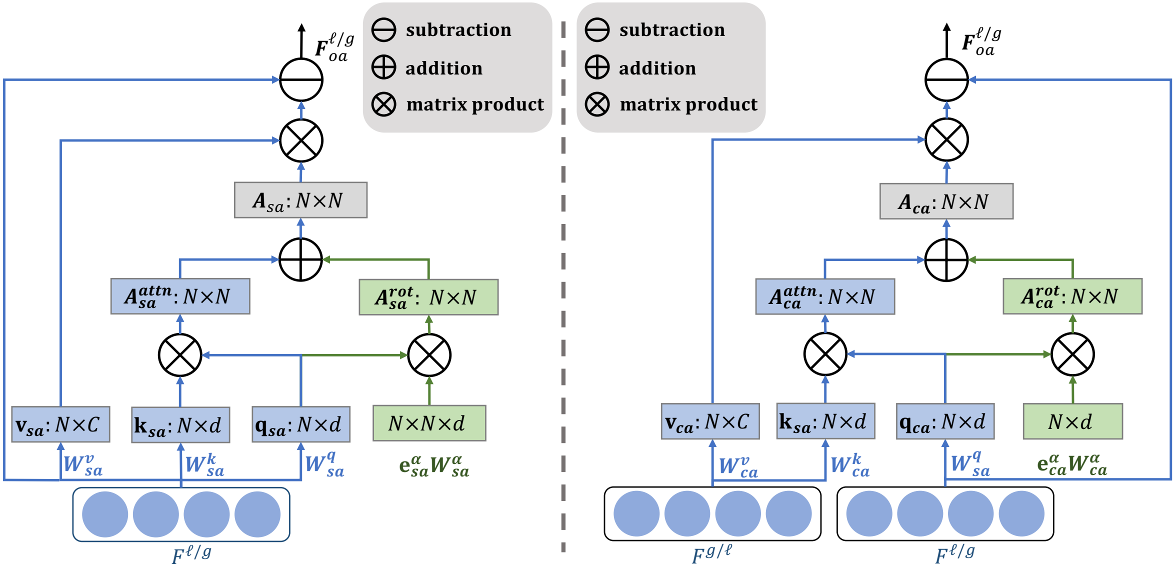
$$\vec{x}^g = \begin{cases} \vec{x}^g, & \text{if } S_x \geq \frac{N_0}{2} \\ \vec{x}'^g, & \text{otherwise} \end{cases}, \quad S_x = \sum_{i=1}^{N_0} \mathbb{1}[\langle \vec{x}^g, \overrightarrow{p_m p_i} \rangle], \tag{3.3}$$

where $\langle \cdot, \cdot \rangle$ is the inner product, $\mathbb{1}[\cdot]$ is a binary indicator that returns 1 if the input argument is positive, otherwise 0. S_x denotes the number of points where $\vec{x}^g$ and $\overrightarrow{p_m p_i}$ point to the same direction. The same rule is applied to disambiguate $\vec{y}^g$ and $\vec{z}^g$ by S_y and S_z. Besides, as mentioned in [114], $\mathbf{M}^g$ might be non-rotational (e.g., reflection). To ensure $\mathbf{M}^g$ a valid rotation, we simply reverse the direction of the basis vector whose S value is the smallest.

3.3.2 Aligned Feature Integration

Transformer has been widely used in 3D domain to capture long-range dependencies [255]. In this section, we introduce Aligned Integration Transformer (AIT), an effective transformer to align latent shape codes with relative rotation angles and integrate information via attention-based integration [36]. Within each AIT module, we first apply Intra-frame Aligned Self-attention on $\mathbf{F}^\ell$ and we do not encode $\mathbf{F}^g$, which is treated as supplementary information to assist local geometry learning with the global shape overview. We discuss that encoding $\mathbf{F}^g$ via self-attention can increase model overfitting, thus lowering the model performance. We will validate our discussion in Section 3.4.7. Inter-frame Aligned Cross-attention is applied on both $\mathbf{F}^\ell$ and $\mathbf{F}^g$, and we use Attention-based Feature Integration module for information Aggregation.

Intra-frame Aligned Self-attention. Point-wise features of $\mathbf{F}^\ell$ are encoded from *unaligned* LRFs, so direct implementation of self-attention on $\mathbf{F}^\ell$ can cause feature inconsistency during integration. To solve this problem, rigid transformations between distinct LRFs are considered, which are explicitly encoded and injected into point-wise relation learning process. We begin by understanding the transformation between two LRFs. For any pair of local orthonormal bases $\mathbf{M}^\ell_i$ and $\mathbf{M}^\ell_j$, a rotation can be easily derived $\Delta\mathbf{R}_{ji} = \mathbf{M}^\ell_i \mathbf{M}^{\ell\top}_j$ and translation is defined as $\Delta\mathbf{t}_{ji} = o^\ell_i - o^\ell_j$, where $o^\ell_{i/j}$ indicates the origin.


⊖ subtraction
⊕ addition
⊗ matrix product
$F^{\ell/g}_{oa}$
$A_{sa}: N\times N$
$A^{attn}_{sa}: N\times N$
$A^{rot}_{sa}: N\times N$
$\mathbf{v}_{sa}: N\times C$
$\mathbf{k}_{sa}: N\times d$
$\mathbf{q}_{sa}: N\times d$
$N\times N\times d$
W^{v}_{sa}
W^{k}_{sa}
W^{q}_{sa}
$\mathbf{e}^{\alpha}_{sa}W^{\alpha}_{sa}$
$F^{\ell/g}$
$A_{ca}: N\times N$
$A^{attn}_{ca}: N\times N$
$A^{rot}_{ca}: N\times N$
$\mathbf{v}_{ca}: N\times C$
$\mathbf{k}_{sa}: N\times d$
$\mathbf{q}_{ca}: N\times d$
$N\times d$
W^{v}_{ca}
W^{k}_{ca}
W^{q}_{sa}
$\mathbf{e}^{\alpha}_{ca}W^{\alpha}_{ca}$
$F^{g/\ell}$
$F^{\ell/g}$

(a) Intra-frame Aligned Self-attention **(b) Inter-frame Aligned Cross-attention**

FIGURE 3.3. Illustrations of (a) Intra-frame Aligned Self-attention and (b) Inter-frame Aligned Cross-attention modules. Note that we only present processes for computing $\mathbf{F}_{oa}$ in both modules.

Although $\Delta\mathbf{R}_{ji}$ is invariant to rotations, we do not directly project it into the embedding space, as it is sensitive to the order of matrix product: $\Delta\mathbf{R}_{ji} \neq \Delta\mathbf{R}_{ij}$, giving inconsistent rotation information when the product order is not maintained. To address this issue, we construct our embedding via the relative rotation angle $\Delta\alpha_{ji}$ between $\mathbf{M}^{\ell}_{i}$ and $\mathbf{M}^{\ell}_{j}$, which is normally used in most PCR works [251, 160] for evaluations. The relative rotation angle $\Delta\alpha_{ji}$ is computed as:

$$\Delta\alpha_{ji} = \arccos\left(\frac{\mathrm{Trace}\left(\Delta\mathbf{R}_{ji}\right) - 1}{2}\right)\frac{180}{\pi} \in [0, \pi], \tag{3.4}$$

where it is easy to see that $\Delta\alpha_{ji} = \Delta\alpha_{ij}$. We further apply sinusoidal functions on $\Delta\alpha_{ji}$ to generate N^2 pairs of angular embeddings $\mathbf{e}^{\alpha} \in \mathbb{R}^{N\times N\times d}$ for all N points as:

$$e^{\alpha}_{i,j,2k} = \sin\left(\frac{\Delta\alpha_{ji}/t_{\alpha}}{10000^{2k/d}}\right), \; e^{\alpha}_{i,j,2k+1} = \cos\left(\frac{\Delta\alpha_{ji}/t_{\alpha}}{10000^{2k/d}}\right), \tag{3.5}$$

where t_{α} controls the sensitivity to angle variations.

Finally, we inject $\mathbf{e}^{\alpha}$ into offset attention and learn intra-frame aligned feature $\mathbf{F}^{\ell}_{\text{IAS}}$ via self-attention as follows:

$$
\begin{aligned}
\mathbf{F}^{\ell}_{\text{IAS}} = \phi\left(\mathbf{F}^{\ell}_{oa}\right) + \mathbf{F}^{\ell},\ \mathbf{F}^{\ell}_{oa} &= \mathbf{F}^{\ell} - \|\,\mathrm{SM}(\mathbf{A}_{sa})\|_1 \mathbf{v}_{sa}, \\
\mathbf{A}_{sa} &= \mathbf{A}^{attn}_{sa} + \mathbf{A}^{rot}_{sa}, \\
\mathbf{A}^{attn}_{sa} = \mathbf{q}_{sa}\mathbf{k}^{\top}_{sa},\ \mathbf{A}^{rot}_{sa} &= \mathbf{q}_{sa}(\mathbf{e}^{\alpha}_{sa}\mathbf{W}^{\alpha}_{sa})^{\top},
\end{aligned}
\tag{3.6}
$$

where $\mathbf{q}_{sa}/\mathbf{k}_{sa}/\mathbf{v}_{sa} = \mathbf{F}^{\ell}\mathbf{W}^{\mathbf{q}}_{sa}/\mathbf{F}^{l}\mathbf{W}^{\mathbf{k}}_{sa}/\mathbf{F}^{l}\mathbf{W}^{\mathbf{v}}_{sa}$, $\mathbf{W}^{\alpha}_{sa} \in \mathbb{R}^{d\times d}$ is a linear projection to refine the learning of $\mathbf{e}^{\alpha}_{sa}$, and $\mathbf{A}_{sa}$ is the attention logits. The same process can be performed for $\mathbf{F}^{g}$ by swapping the index ℓ and g. Detailed illustrations are shown in Figure 3.3 (a).

Inter-frame Aligned Cross-attention. Semantic information exchange between $\mathbf{F}^{\ell}$ and $\mathbf{F}^{g}$ in the feature space is implemented efficiently by cross-attention [29]. Since $\mathbf{F}^{\ell}$ and $\mathbf{F}^{g}$ are learned from different coordinate systems, inter-frame transformations should be considered for cross-consistency between $\mathbf{F}^{\ell}$ and $\mathbf{F}^{g}$. An illustration of the cross-attention module is shown in Figure 3.3 (b), which indicates that the computation of inter-frame aligned feature $\mathbf{F}^{\ell}_{\text{IAC}}$ via cross-attention follows a similar way as Eq. 3.6 by replacing all subscripts sa by ca. As illustrated in Figure 3.3 (b), $\mathbf{A}_{ca}$ is cross-attention logits containing point-wise cross-relations over point features defined across local and global scales. $\mathbf{e}^{\alpha}_{ca} \in \mathbb{R}^{N\times d}$ is computed via Eq. 3.4 and Eq. 3.5 in terms of the transformation between M^{ℓ}_{i} and M^{g}. To this end, the geometric features learned between local and global reference frames can be aligned given $\mathbf{e}^{\alpha}_{ca}$, leading to a consistent feature representation.

Attention-based Feature Integration. Instead of simply adding the information from both $\mathbf{F}^{\ell}$ and $\mathbf{F}^{g}$, we integrate information by incrementing attention logits. Specifically, we apply self-attention on $\mathbf{F}^{\ell}$ with attention logits $\mathbf{A}_{sa}$ and cross-attention between $\mathbf{F}^{\ell}$ and $\mathbf{F}^{g}$ with attention logits $\mathbf{A}_{ca}$. We combine $\mathbf{A}_{sa}$ and $\mathbf{A}_{ca}$ via addition, so that encoded information of all point pairs from a local domain can be enriched by the global context of the whole shape. The whole process is formulated as follows:

$$
\begin{aligned}
\mathbf{U} &= \phi\left(\mathbf{F}_{oa}\right) + \mathbf{F}^{\ell}, \\
\mathbf{F}_{oa} &= \mathbf{F}^{\ell} - \|\,\mathrm{SM}(\mathbf{A}_{sa} + \mathbf{A}_{ca})\|_1(\mathbf{v}_{sa} + \mathbf{v}_{ca}).
\end{aligned}
\tag{3.7}
$$

Hence, intra-frame point relations can be compensated by inter-frame information communication in a local-to-global manner, which enriches the geometric representations.

3.3.3 Deep Feature Registration

Correspondence mapping [221, 160] plays an important role in PCR, and we discuss that it is also critical for achieving RI in our design. Specifically, although $\mathbf{F}^{\ell}$ and $\mathbf{F}^{g}$ are both rotation-invariant by theory, different point sampling methods and the sign ambiguity will cause the final feature not strictly rotation-invariant. To solve this issue, we first examine the correspondence map:

$$m\left(\mathcal{X}, \mathcal{Y}\right) = \frac{\exp\left(\varPhi_1(\mathcal{Y})\varPhi_2(\mathcal{X})^\top/t\right)}{\sum_{j=1}^{N}\exp\left(\varPhi_1(\mathcal{Y})\varPhi_2(\boldsymbol{x}_j)^\top/t\right)}, \tag{3.8}$$

where $\varPhi_1$ and $\varPhi_2$ are MLPs that project latent embeddings $\mathcal{X}$ and $\mathcal{Y}$ to a shared space, and t controls the variation sensitivity. It can be seen from Eq. 3.8 that the mapping function m reveals feature similarities in the latent space, and it is also an essential part for 3D point-level contrastive learning in PointContrast [238] for the design of InfoNCE losses [207], which have been proven to be equivalent to maximize the mutual information. Based on this observation, we propose a registration loss function $\mathcal{L}_r = \mathcal{L}_r^{\ell} + \mathcal{L}_r^{g}$, where $\mathcal{L}_r^{\ell}$ and $\mathcal{L}_r^{g}$ represent the registration loss of ($\mathbf{F}^{\ell}$,$\mathbf{U}$) and ($\mathbf{F}^{g}$,$\mathbf{U}$). Mathematically, $\mathcal{L}_r^{\ell}$ is defined as follows:

$$\mathcal{L}_r^{\ell} = -\sum_{(i,j)\in M}\log\frac{\exp\left(\varPhi_1(\mathbf{U}_j)\varPhi_2(\mathbf{f}_i^{\ell})^\top/t\right)}{\sum_{(\cdot,k)\in M}\exp\left(\varPhi_1(\mathbf{U}_k)\varPhi_2(\mathbf{f}_i^{\ell})^\top/t\right)}. \tag{3.9}$$

The same rule is followed to compute $\mathcal{L}_r^{g}$. Although we follow the core idea of PointContrast, we differ from it in that PointContrast defines positive samples based on feature correspondences computed at the same layer level, while our positive samples are defined across layers.

The intuition for the loss design is that the 3D shape is forced to learn about its local region as it has to distinguish it from other parts of different objects. Moreover, we would like to maximize the mutual information between different poses of the 3D shape, as features encoded from different poses should represent the same object, which is very useful in achieving RI in

SO(3). Moreover, the mutual information between $\mathbf{F}^{\ell}$ and $\mathbf{F}^{g}$ is implicitly maximized, such that shared semantic information about geometric structures can be learned, leading to a more geometrically accurate and discriminative representation.

3.4 Experiments and Results

We evaluate our model on 3D shape classification, part segmentation, and retrieval tasks under rotations, and extensive experiments are conducted to analyze the network design. We follow [59] for evaluation: training and testing the network under z-axis (z/z); training under z-axis and testing under arbitrary rotations (z/SO(3)); and training and testing under arbitrary rotations (SO(3)/SO(3)).

3.4.1 Datasets

In this work, we use two datasets for the classification task: ModelNet40 [233] and ScanObjectNN [205], one dataset for the part segmentation task: ShapeNetPart [252], and one dataset for the shape retrieval task: ShapeNetCore55 [27]. Please refer to Section 2.4.1 for the details of ModelNet40 and ShapeNetPart.

ScanObjectNN. ScanObjectNN consists of 3D point clouds of objects segmented from the ScanNet scenes [46], which is particularly suited for benchmarking 3D point cloud classification task in real-world scenario. It contains around 15,000 objects that are categorized into 15 categories with 2902 unique object instances. Detailed samples of ScanObjectNN are present in Figure 3.4.

ShapeNetCore55. ShapeNetCore55 contains a selection of categories from the ShapeNet dataset, with each category containing a number of 3D models, which is suitable for benchmarking 3D shape retrieval tasks, including two categories of datasets: normal and perturbed. Detailed samples of ShapeNetCore55 are shown in Figure 3.5.

FIGURE 3.4. Visualization of ScanObjectNN dataset.

FIGURE 3.5. Visualization of ShapeNetCore55 dataset.

3.4.2 Evaluation Metrics

For the classification and segmentation tasks, we follow the evaluation metrics mentioned in Section 2.4.1 by using the accuracy score and mIoU respectively to evaluate performances of different methods. For the shape retrieval task, we use the common metric to evaluate the performance: mean Average Precision (mAP). The mathematical expression for mAP is:

$$\begin{aligned} \text{mAP} &= \frac{1}{Q}\sum_{q=1}^{Q}\text{AP}(q), \\ \text{AP}(q) &= \frac{1}{\text{number of relevant items for } q}\sum_{k=1}^{N}P(k)\times\delta(k), \\ R(k) &= \frac{\text{number of relevant items among the top } k \text{ retrieved items}}{\text{total number of relevant items}}, \\ P(k) &= \frac{\text{number of relevant items among the top } k \text{ retrieved items}}{k}, \end{aligned} \tag{3.10}$$

where $P(k)$ denotes precision, indicating the proportion of true positive retrievals to the total retrieved items within the top k selected items. $R(k)$ denotes recall, the proportion of true positive retrievals to the total number of actual relevant items within the top k selected items. AP(q) denotes Average Precision, which computes the average precision values at the ranks where a relevant item is retrieved. $\delta(k)$ is an indicator function that is 1 if the item at rank k is relevant and 0 otherwise/

3.4.3 Implementation Details

For all three tasks, we set the batch size to 32 for training and 16 for testing. We use farthest point sampling to re-sample the points from the initial 10k points to 1024 points for classification and retrieval and 2048 points for segmentation. Random point translation within $[-0.2, 0.2]$ and re-scaling within $[0.67, 1.5]$ were adopted for augmentation. We trained the model for 250 epochs with t_α = 15 and t = 0.017. SGD is adopted as the optimizer, where the learning rate was set to 1e-2 with momentum of 0.9 and weight decay of 1e-4. Cosine annealing was applied to reschedule the learning rate for each epoch. For classification and retrieval, we used one RTX2080Ti GPU with PyTorch for model implementation, and we used two GPUs for the segmentation task. The normal vector information is ignored for all experiments.

3.4.4 3D Object Classification

Synthetic Dataset. We first examine the model performance on the synthetic ModelNet40 [233] dataset. We sample 1024 points from each data with only xyz coordinates as input features. Hyper-parameters for training follow the same as [66], except that points are downsampled in the order of (1024, 512, 128) with feature dimensions of (3, 128, 256). We report and compare our model performance with state-of-the-art (SoTA) methods in Table 3.1. Both rotation sensitive and robust methods achieve great performance under z/z. However, the former could not generalize well to unseen rotations. Rotation robust methods like SFCNN [172] achieve competitive results under z/z, but their performance is not consistent on z/SO(3)

TABLE 3.1. Classification results on ModelNet40. All methods take raw points of 1024×3 as inputs.

Rotation Sensitive	z/z	z/SO(3)	SO(3)/SO(3)
PointNet [164]	89.2	16.2	75.5
PoinNet++ [166]	89.3	28.6	85.0
PCT [66]	90.3	37.2	88.5
Rotation Robust	z/z	z/SO(3)	SO(3)/SO(3)
SFCNN [172]	**91.4**	84.8	90.1
RIConv [263]	86.5	86.4	86.4
SRINet [195]	87.0	87.0	87.0
ClusterNet [28]	87.1	87.1	87.1
PR-InvNet [256]	89.2	89.2	89.2
RI-GCN [103]	89.5	89.5	89.5
GCAConv [262]	89.0	89.1	89.2
RI-Framework [116]	89.4	89.4	89.3
VN-DGCNN [49]	89.5	89.5	90.2
SGMNet [239]	90.0	90.0	90.0
[114]	90.2	90.2	90.2
OrientedMP [138]	88.4	88.4	88.9
ELGANet [65]	90.3	90.3	90.3
Ours	91.0	**91.0**	**91.0**

TABLE 3.2. Classification results on ScanObjectNN OBJ_BG.

Method	z/SO(3)	SO(3)/SO(3)
PointNet [164]	16.7	54.7
PointNet++ [166]	15.0	47.4
DGCNN [223]	17.7	71.8
PCT [66]	28.5	45.8
RIConv [263]	78.4	78.1
RI-GCN [103]	80.5	80.6
GCAConv [262]	80.1	80.3
RI-Framework [116]	79.8	79.9
LGR-Net [264]	81.2	81.4
VN-DGCNN [49]	79.8	80.3
OrientedMP [138]	76.7	77.2
Ours	**86.6**	**86.3**

and SO(3)/SO(3) due to the imperfect projection from points to voxels when using spherical solutions. We outperform the recent proposed methods [138, 239, 49] and achieve an accuracy of 91.0%, proving the superiority of our framework on classification.

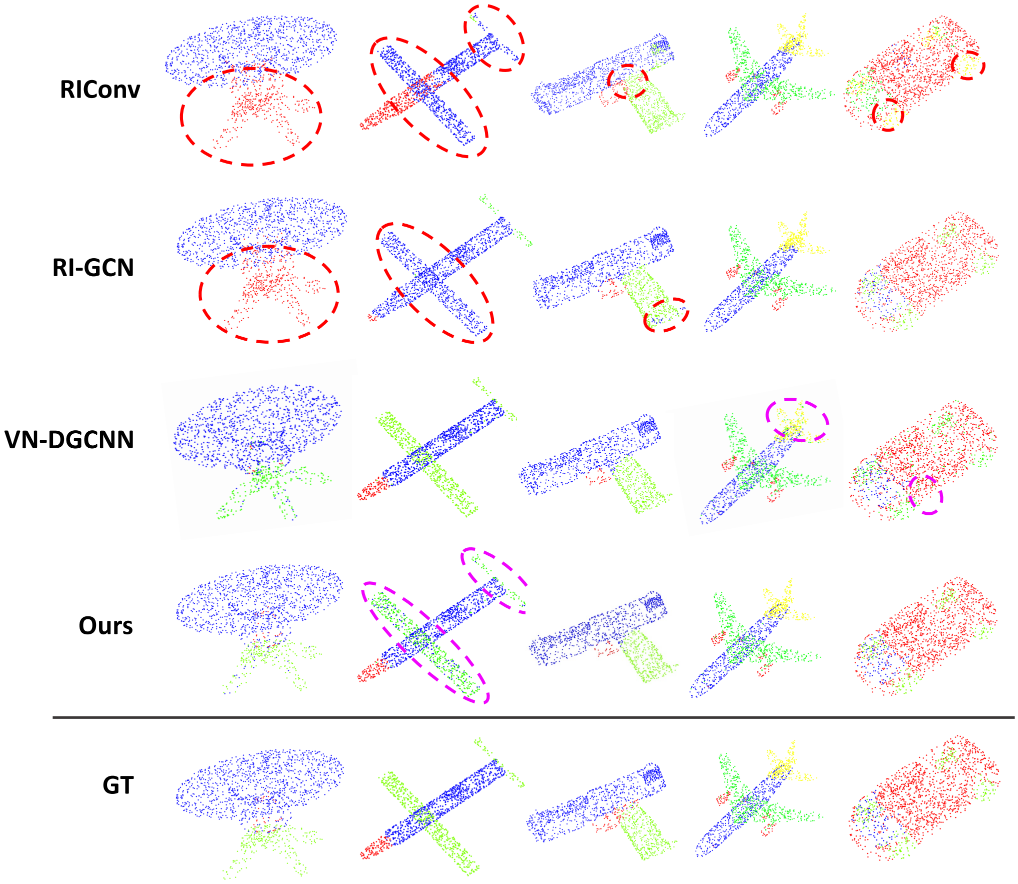


FIGURE 3.6. Segmentation comparisons on ShapeNetPart, where ground-truth samples are shown for reference. Red dotted circles indicate obvious failures on certain classes, and purple circles denote the slight difference between our design and VN-DGCNN.

TABLE 3.3. Segmentation results on ShapeNetPart. The second best results are underlined.

Method	z/SO(3)	SO(3)/SO(3)
PointNet [164]	38.0	62.3
PointNet++ [166]	48.3	76.7
PCT [66]	38.5	75.2
RIConv [263]	75.3	75.5
RI-GCN [103]	77.2	77.3
RI-Framework [116]	79.2	79.4
LGR-Net [264]	80.0	80.1
VN-DGCNN [49]	**81.4**	**81.4**
OrientedMP [138]	80.1	<u>80.9</u>
Ours	<u>80.3</u>	80.4

Real Dataset. Experiments are also conducted on a real-scanned dataset, i.e., ScanObjectNN. We use *OBJ_BG* subset with the background noise and sample 1,024 points under z/SO(3) and SO(3)/SO(3). Table 3.2 shows that our model achieves the highest results with excellent consistency with random rotations.

3.4.5 3D Part Segmentation

Shape part segmentation is a more challenging task than object classification. We use ShapeNetPart [252] for evaluation, where we sample 2048 points with xyz coordinates as model inputs. The training strategy is the same as the classification task except that the training epoch number is 300. Representative methods such as PointNet++ and PCT are vulnerable to rotations. Rotation robust methods present competitive results under z/SO(3), where we achieve the second best result of 80.3$\%$. Moreover, qualitative results shown in Figure 3.6 present that we can achieve visually better results than VN-DGCNN in certain classes such as the airplane and car.

3.4.6 3D Shape Retrieval

We further conduct 3D shape retrieval experiments on ShapeNetCore55 [27]. We only use the perturbed part to validate our model performance under rotations. We combine the training and validation sets and validate our method on the testing set following the training policy of [59]. Experimental results are reported in Table 3.4, where the final score is the average value of micro and macro mean average of precision (mAP) as in [184]. Similar to the classification task, our method achieves SoTA performance.

3.4.7 Ablation Study

3D Semantic Segmentation. To check our model's effectiveness on real-world large scenes, additional experiments are conducted on S3DIS dataset [10], which includes six indoor areas of three different buildings. Each point is labeled by one of the 13 categories (e.g., ceiling, chair or clutter). Following the same pre-processing steps as [166, 223], each room is divided

TABLE 3.4. Comparisons of SoTA methods on the 3D shape retrieval task.

Method	micro mAP	macro mAP	Score
Spherical CNN [59]	0.685	0.444	0.565
SFCNN [172]	0.705	0.483	0.594
GCAConv [262]	0.708	0.490	0.599
RI-Framework [116]	0.707	**0.510**	0.609
Ours	**0.715**	**0.510**	**0.613**

TABLE 3.5. Semantic segmentation results (mIoU) on S3DIS area-5.

Method	z/z	z/SO(3)	SO(3)/SO(3)
PointNet [164]	41.1	4.1	29.3
DGCNN [223]	48.4	3.6	34.3
RIConv [263]	22.0	22.0	22.0
LRG-Net [264]	43.4	43.4	43.4
Ours	**51.2**	**51.2**	**51.2**

TABLE 3.6. Module analysis of AIT and loss functions. $\mathbf{F}^{g*}$ means encoding $\mathbf{F}^{g}$ via Intra-frame Aligned Self-attention.

Model	$\mathbf{e}_{sa}^{\alpha}$	$\mathbf{e}_{ca}^{\alpha}$	$\mathbf{F}^{g*}$	$\mathbf{A}_{sa} + \mathbf{A}_{ca}$	$\mathcal{L}_r^{\ell}$	$\mathcal{L}_r^{g}$	Acc.
A				✓	✓	✓	90.0
B	✓			✓	✓	✓	90.6
C		✓		✓	✓	✓	90.2
D	✓	✓	✓	✓	✓	✓	90.2
E	✓	✓			✓	✓	90.4
F	✓	✓		✓			90.0
G	✓	✓		✓	✓		90.2
H	✓	✓		✓		✓	90.6
Ours	✓	✓		✓	✓	✓	**91.0**

into 1m×1m blocks and for each block 4096 points are sampled during training process. We use area-5 for testing and all the other areas for training. The quantitative results are shown in Table 3.5 following [264], where it shows that under random rotations, our model outperforms LGR-Net by 7.8%, showing a more effective way to process large indoor scenes. For a more intuitive understanding of our model performance, qualitative results are shown in Figure 3.7 for reference.

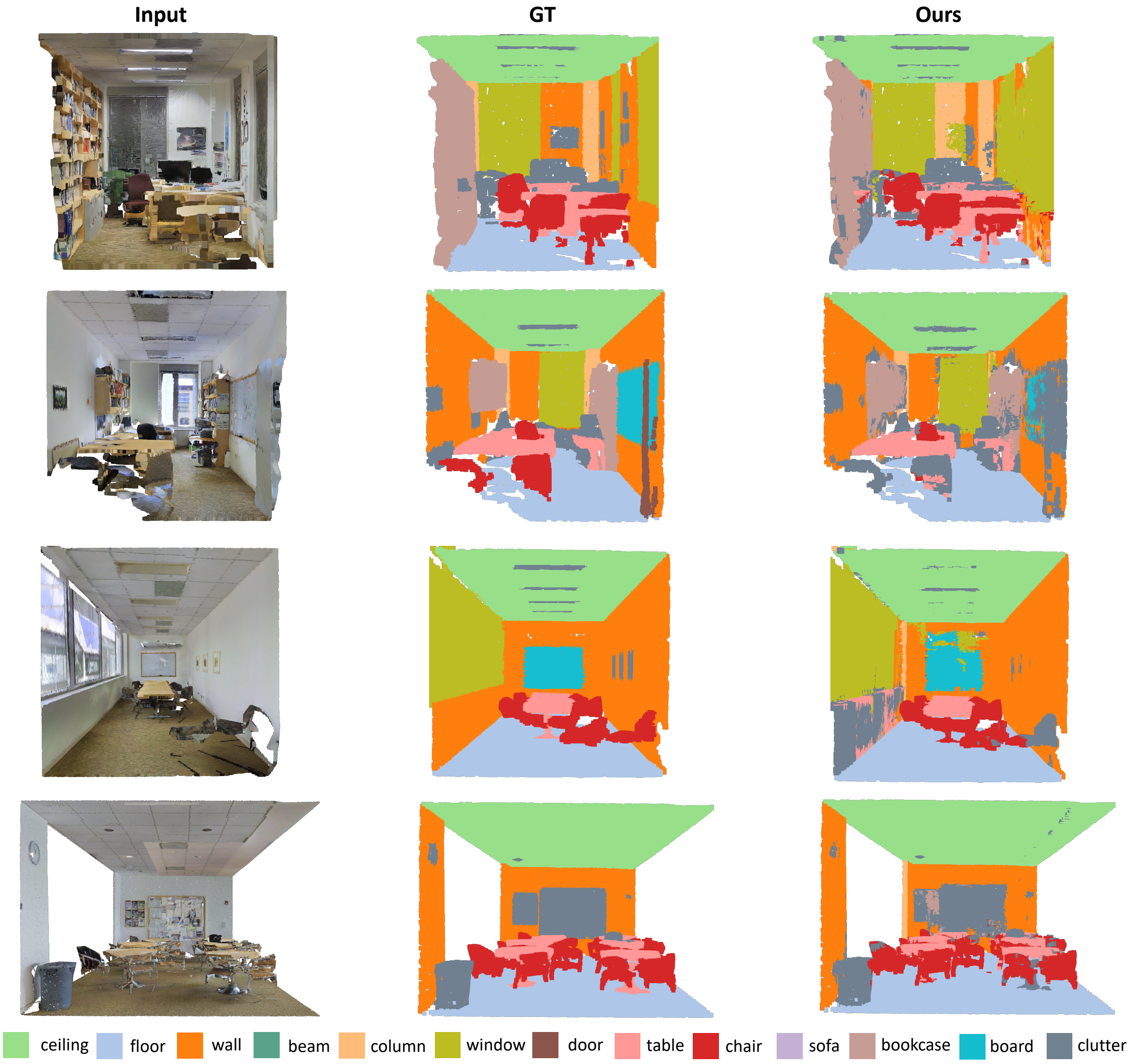


FIGURE 3.7. Visualization of semantic segmentation results on S3DIS area-5. The first row is the original inputs, the second row is the ground-truth samples and the last row is our predicted results.

Effectiveness of Transformer Designs. We examine the effectiveness of our transformer design by conducting classification experiments under z/SO(3). We first ablate one or both of the angular embeddings and report the results in Table 3.6 (models A, B, and C). Model B performs better than model C by 0.4%, which validates our design of feature integration where M_i^ℓ is used as the main source of information. When both angular embeddings are applied, the best result is achieved (i.e., 91.0%). Moreover, we validate our discussion in Section 3.3.2 by comparing models D and E. We demonstrate in model D that when encoding

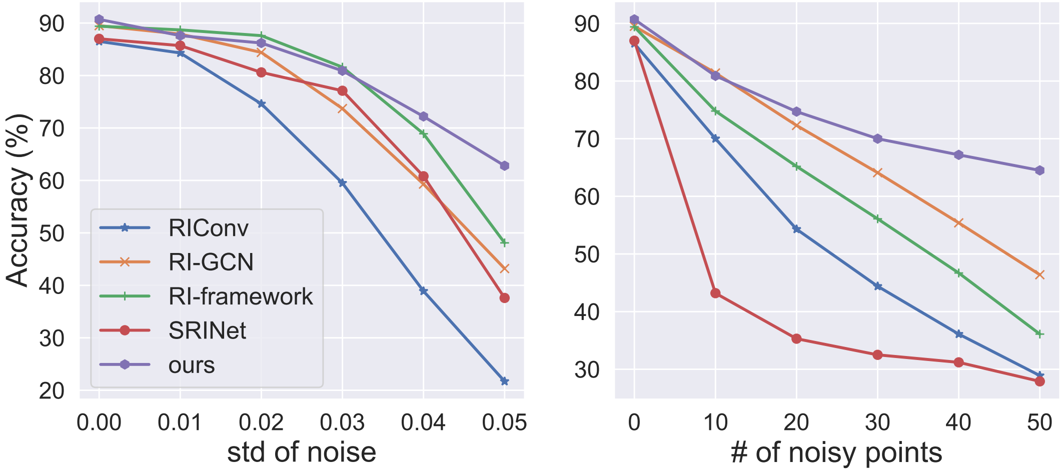


Figure 3.8. Left: Results on Gaussian noise of zero mean and variant standard deviation values. Right: Results on different numbers of noisy points.

$\mathbf{F}^g$ in the same way as $\mathbf{F}^\ell$, the model performance decreases, which indicates that encoding $\mathbf{F}^g$ via self-attention will increase the model overfitting. Finally, we examine the effectiveness of our attention logits-based integration scheme by comparing our model with the conventional method (model E), which applies self- and cross-attention sequentially and repeatedly. We observe that our result is better than model E by 0.6%, indicating that our design is more effective.

Registration Loss. We sequentially ablate $\mathcal{L}_r^g$ and $\mathcal{L}_r^\ell$ (models F, G, and H) to check the effectiveness of our registration loss deign. Results in Table 3.6 demonstrate that we can still achieve a satisfactory result of 90.0% without feature registration. Individual application of $\mathcal{L}_r^g$ and $\mathcal{L}_r^\ell$ shows the improvement when forcing the final representation to be close to rotation-invariant features. Moreover, it can be seen that model H performs better than model G, which indicates that intermediate features learned from the global scale are important for shape classification. The best model performance is hence achieved by applying both losses.

Noise Robustness. In real-world applications, raw point clouds contain noisy signals. We conduct experiments to present the model robustness to noise under z/SO(3). Two experiments are conducted: (1) We sample and add Gaussian noise of zero mean and varying standard deviations $\mathcal{N}(0, \sigma^2)$ to the input data; (2) We add outliers sampled from a unit sphere to each object. As shown in Figure 3.8 (left), we achieve on par results to RI-Framework when std is

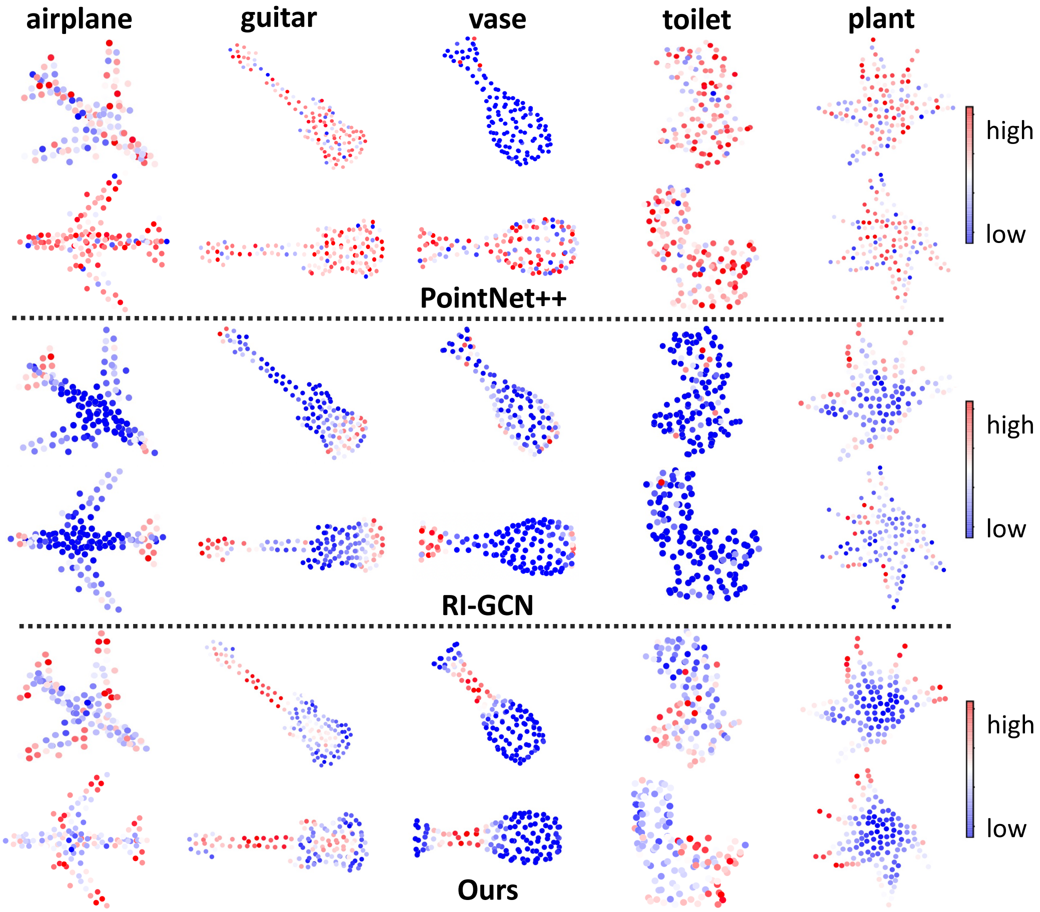


FIGURE 3.9. Network attention on PointNet++, RI-GCN and our model.

low, while we perform better while std increases, indicating that our model is robust against high levels of noise. Besides, as the number of noisy points increases, most methods are heavily affected while we can still achieve good results.

Visualization of Rotation Invariance. We further examine RI of learned features. Specifically, we use Grad-CAM [187] to check how the model pays attention to different parts of data samples under different rotations. Results are reported in Figure 3.9 with correspondence between gradients and colors shown on the right. RI-GCN presents a good result, but its behavior is not consistent over some classes (e.g., vase and plant) and it does not pay attention to regions that are critical for classification (see toilet), showing inferior performance to ours. PointNet++ shows no resistance to rotations, while our method exhibits a consistent gradient

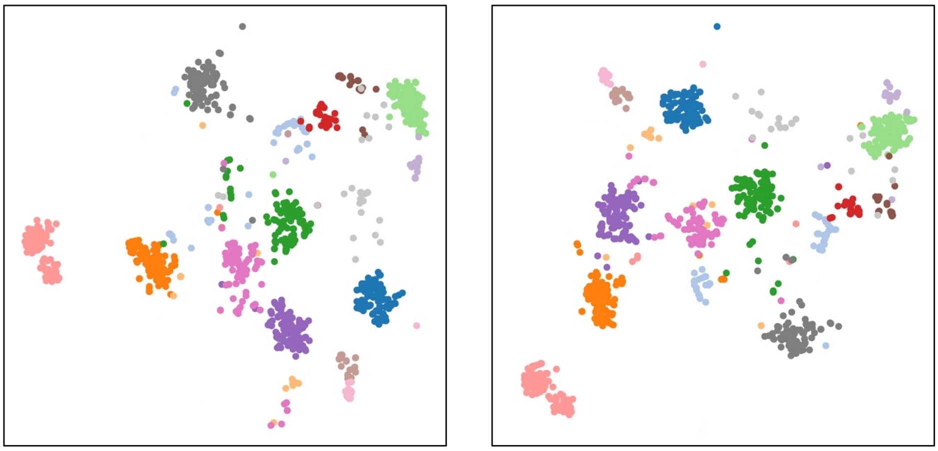


FIGURE 3.10. t-SNE of the aggregated **U** with z/SO(3) (**Left**) and SO(3)/SO(3) (**Right**). Clusters indicate good predictions in object classification.

distribution over different parts with random rotations, indicating our network is not affected by rotations.

Visualization of U. To better present the discriminability of the learned features, we summarize the shape feature representation **U** by maxpooling and visualize it via t-SNE [208]. Experiments are conducted on object classification under z/z and z/SO(3). Only the first 16 classes are selected for a clear representation purpose as shown in Figure 3.10. Although it is difficult to correctly separate all categories, we can see that some shape classes can be perfectly predicted, and the overall representation ability of **U** under different testing protocols is satisfactory and consistent.

3.5 Discussion

We have demonstrated in Section 3.4 that our method is robust to the white noise and number of points, but we have also examined that our method can be sensitive to randomness and the implementation is not quite efficient. In the following, we conduct two additional experiments on model randomness and model complexity and discuss our model performances under these two different circumstances.

TABLE 3.7. Variance and Mean values of different model performances on ModelNet40 with z/SO(3).

Model	A	B	C
Acc. (%)	89.8±0.2	90.4±0.2	90.1±0.1
Model	D	E	F
Acc. (%)	89.8±0.4	90.1± 0.3	89.6±0.4
Model	G	H	Best
Acc. (%)	90.0±0.2	90.3±0.3	90.8±0.4

Influence of Randomness. We first examine the robustness of our model to 3D point clouds of random poses. The mean and variance values of performances are reported in Table 3.7, where the total training epochs remain the same as the previous designs. We can see that although our best model can achieve the best performance of 90.8%, it is quite sensitive to point clouds of different poses with a variance of 0.4%. When compared to other designs such as models A, B, and C, they all have a relatively lower model variance than our best model, which indicate that it is hard for our best model to achieve a stable performance when given different raw point clouds.

Model Complexity. Inference model sizes of different methods along with the corresponding construction time for LRFs and inference speed are reported in Table 3.8. The construction time measured in seconds shows time cost for different models generating their low-level rotation-invariant shape features, where we record the total time for local and global representation constructions of RI-Framework and our work. As can been seen, our proposed method takes a relatively longer time than most methods, and our model parameters is the largest amongst all compared methods, indicating that further designs are needed to solve this

TABLE 3.8. Model complexity construction time for LRFs, and inference speed on ModelNet40 with z/SO(3), where [114] is considered without test time augmentation.

Method	Params (M)	Times (s)	Speed (ins./s)	Acc (%)
RIConv [263]	0.68	0.041	396.4	86.4
RI-GCN [103]	4.19	0.057	139.1	89.5
RI-Framework [116]	2.36	0.134	43.1	89.4
VN-DGCNN [49]	2.77	-	77.3	89.5
Li et al. [114]	2.76	0.047	35.8	90.2
Ours	3.11	0.043	205.3	91.0

issue, and the trade-off between the accuracy and inference speed is hard to balance. We will investigate the model design for a much high accuracy and faster speeds in the future work.

3.6 Summary

In this chapter, we rethink and investigate the close relation between rotation invariance and point cloud registration, based on which we propose a PCR-cored learning framework with three stages. With a pair of rotation-invariant shape descriptors constructed from local and global scales, a comprehensive learning and feature integration module is proposed, Aligned Integration Transformer, to simultaneously effectively align and integrate shape codes via self- and cross-attentions. To further preserve rotation invariance in the final feature representation, a registration loss is proposed to align it with intermediate features, where shared semantic knowledge of geometric parts is also extracted. Extensive experiments demonstrated the superiority and robustness of our designs.

In addition, the designed PRC-cored learning framework can be applied to large-scale point cloud datasets, and we have demonstrated the ability of our model in chapter 3 when we apply it to semantic3d dataset, which is a large-scale point cloud classification benchmark. The current challenge to consider is that due to the use of attention mechanism, the computational cost could be quite high when more points are handled, which leads to a large model. Moreover, based on study of [139], it is easy for a large model to forget the knowledge that it has learned at the early stage, so this also introduces another difficulty. In future work, we will examine efficient methods for invariance learning on large-scale point clouds.

CHAPTER 4

Deep Learning-based Modeling for 3D Human Mesh Texturing

This chapter concentrates on the realm of 3D human meshes, which presents numerous intriguing research prospects and advantages, primarily due to the pivotal role of human models across multiple domains. For example, in the computer graphics field, human meshes can foster more precise and lifelike character models and animations [213]. This emphasis on human meshes not only bolsters our understanding of the human form, but also enriches the practical applications of 3D modeling and interaction. However, a large number of human mesh datasets, such as Caesar [88] and ScanDB [71], ignore human mesh textures during data collection stage, which heavily hinders the progress in the human texturing domain. Hence, it is an urgent need to develop an automatic generation pipeline for the human mesh texturing. In this chapter, we focus on text-guided human texture generation using deep learning-based methods, and we reveal our analysis and investigations on human mesh generation.

4.1 Introduction

The task of developing 3D human avatars from text or visual cues has been a complex and difficult problem in the fields of 3D data analysis. Previous methods have often relied on intricate and costly equipment to create high-quality avatar models. Nevertheless, these techniques necessitate the use of multi-perspective images or depth maps, which are typically beyond the budget of consumer-level applications. Alternatively, some works [181, 182] make use of neural networks to estimate viable avatar models from a single image. However, these methods are constrained by the accessibility of appropriate images and they lack the capability to be modified once an image has been selected as a reference. Significant progress has been made in text-to-3D shape generation. Some methods are proposed for general objects [162, 121], and some are specifically for 3D human avatars [24, 83, 91]. Success of these methods rely on text-to-image generation, which leverages the diffusion model [80, 176], and Score Distillation Sampling (SDS) [162] combined with differentiable 3D representations [147, 14].

In our study, we utilize 3D scene representations that are more conducive to generating high-quality 3D human textures from textual descriptions. We firstly go through some existing methods on text-guided human mesh and texture generation in Section 4.2, where we propose our main designs based on a current method named Fantasia3D [34], which enables us to generate detailed and high-quality 3D human textures. In Section 4.3, we propose to address a primary issue associated with SDS, which usually causes over-smoothed and low-quality textures. Our main idea is to denoise the unclear gradient direction provided by the SDS loss. We handle this from two points of view. Firstly, we propose *Denoising Score Distillation* (DSD), which introduces a negative gradient component to modify the SDS, which could correct the SDS gradient direction iteratively for detailed and high-quality texture generation. Then, to enable geometry-aware texture generation, we utilize geometric guidance which provides rich details of the mesh surface to guide the DSD precisely, and use spatial-aware texture shading models [98] to guarantee the quality of rendered visual results. Finally, we validate our proposed methods on extensive experiments in Section 4.4.

The contributions of this chapter are summarized as follows:

(1) We introduce Denoising Score Distillation, a diffusion-based denoising score using negative image-text pairs for high-fidelity texture generation aligned to textual descriptions.

(2) We employ semantically aligned 2D depth signals and spatially-aware rendering functions for geometry-aware texture generation and realistic avatar rendering.

(3) Through comprehensive experiments, we prove the efficacy of our method over existing texture generation techniques.

4.2 Literature Review

4.2.1 Score Distillation Sampling

Due to the introduction of SDS and the text-to-image diffusion model, 3D content generation is made possible where neural representations of the 3D content can be differentiable, i.e., NeRF [147] or DMTet [189]. In this part, we will go through the implementation of SDS in full detail. Given an input image $\mathbf{x}$, instead of directly performing the forward/backward diffusion process on $\mathbf{x}$, an image encoder is used first to encode the image to a latent space [176], resulting in a latent code $\mathbf{z}$.

A denoising U-Net ϵ_ϕ with model parameters ϕ is then used to encode and decode the latent code $\mathbf{z}$ with a text embedding y introduced at the bottleneck of the U-net. In addition, a timestep $t \sim \mathcal{U}(0, \mathbf{I})$ is uniformly sampled to add the white Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ into $\mathbf{z}$. The diffusion loss can be mathematically formulated as:

$$\mathcal{L}_{\text{Diff}}(\mathbf{z}, y, t) = w(t) \|\epsilon_\phi(\mathbf{z}_t, y, t) - \epsilon\|_2^2, \tag{4.1}$$

where $w(t)$ is a weighting function that depends on the timestep t, and $\mathbf{z}_t$ refers to the noisy version of $\mathbf{z}$ via an iterative forward diffusion process given by $\mathbf{z}_t = \sqrt{\alpha_t}\mathbf{z} + \sqrt{1 - \alpha_t}\epsilon$, with α_t being the noise scheduler.

To enable high quality generation, classifier-free guidance (CFG) [81] is used that jointly learns text-conditioned and unconditioned models through a guidance scale parameter ω. During inference, the two models are used to denoise the image as follows:

$$\hat{\epsilon}_\phi\left(\mathbf{z}_t, y, t\right) = (1 + \omega)\epsilon_\phi\left(\mathbf{z}_t, y, t\right) - \omega\epsilon_\phi\left(\mathbf{z}_t, t\right). \tag{4.2}$$

Given a differentiable rendering function g_θ with parameters θ, which renders 3D avatar models into 2D images, the gradient of the diffusion loss function with respect to the rendering function parameters θ is:

$$\nabla_\theta \mathcal{L}_{\text{Diff}} = w(t)\left(\hat{\epsilon}_\phi\left(\mathbf{z}_t, y, t\right) - \epsilon\right) \frac{\partial \hat{\epsilon}_\phi(\mathbf{z}_t, y, t)}{\partial \mathbf{z}_t} \frac{\partial \mathbf{z}_t}{\partial \theta}. \tag{4.3}$$

As demonstrated in [162], omitting the U-Net Jacobian term leads to effective gradient for optimizing g_θ, so the final SDS score function is:

$$\nabla_\theta \mathcal{L}_{\text{SDS}} = w(t) \left(\hat{\epsilon}_\phi \left(\mathbf{z}_t, y, t \right) - \epsilon \right) \frac{\partial \mathbf{z}_t}{\partial \theta}. \tag{4.4}$$

The purpose of SDS is to generate samples via optimization from a text-guided diffusion model.

4.2.2 Physically-Based Rendering

Given a textureless mesh with vertex positions, face indices, and texture coordinates, our task is to generate photorealistic surface rendering based on the diffusion model. For this purpose, we choose the Physically-Based Rendering model. Specifically, we apply spatially-varying BRDF and differential rendering functions to make the whole pipeline suitable for realistic texture generation.

Diffuse SV-BRDF. We follow [98] to employ simple diffuse model (i.e., Lambertian or Phong models) for low computational cost. Moreover, to enable differential rendering process, we use a simple multi-layer perceptron $\sigma(\cdot)$ and a positional encoding network $\gamma(\cdot)$ to predict spatially varying albedo term $\mathbf{k}_d = \sigma(\gamma(\mathbf{x}_p)) \in \mathbb{R}^4$ as:

$$f_d(\mathbf{x}_p) = \frac{\mathbf{k}_d}{\pi}. \tag{4.5}$$

Specular SV-BRDF. For the specular part, we use Cook-Torrance microfacet specular shading model [45] to characterize the physical properties of an object's surface. In addition, we employ the material model from [23] for easy rendering:

$$f_s(\mathbf{l}, \mathbf{v}) = \frac{DFG}{4(\mathbf{n} \cdot \mathbf{l})(\mathbf{n} \cdot \mathbf{v})}, \tag{4.6}$$

where $\mathbf{n}$ is the surface normal, $\mathbf{l}$ the input light vector and $\mathbf{v}$ the view direction. The terms D, F and G represent the normal distribution function, the Fresnel term and geometric attenuation. As mentioned in [23] that specular reflectance is related to roughness r and metallic term m, where r serves as a parameter of D and pre-filtered environment map [98]. The metalness term m presents the dielectric and conductor reflectance. The specular reflectance at normal incidence $\mathbf{k}_s = m \cdot \mathbf{k}_d + (1 - m) \cdot 0.04$ serves a parameter of F.

Following the same idea as in diffuse SV-BRDF, we use MLP and positional encoding to predict the specular related terms $(r, m) = \sigma(\gamma(\mathbf{x}_p)) \in \mathbb{R}^2$. Formally, we use the same MLP and positional encoding network to predict diffuse and specular reflectance terms given the surface point $\mathbf{x}_p$: $(\mathbf{k}_d, r, m) = \sigma(\gamma(\mathbf{x}_p)) \in \mathbb{R}^5$.

Rendering. We apply Image-Based Lighting model with the aforementioned SV-BRDFs to render 2D image pixels:

$$R(\mathbf{x}_p, \mathbf{l}) = \int_H L_i(\mathbf{l})(f_d + f_s)\,(\mathbf{l} \cdot \mathbf{n})\,\mathrm{d}\mathbf{l}, \tag{4.7}$$

where $L_i(\mathbf{l})$ is the incident radiance, $H = \{\mathbf{l} : \mathbf{l} \cdot \mathbf{n} \geq 0\}$ demotes a hemisphere with incident light and surface normal. For fast rendering purpose, we employ the differentiable split-sum approximation of Eq. 4.7 and pre-compute a 2D look-up texture map.

4.2.3 3D Shape and Texture Generation

There has been a recent surge of interest in the field of generating 3D shapes and textures. One line of methods, such as Text2Mesh [145], Tango [112], and CLIP-Mesh [152], utilize CLIP-space similarities as an optimization objective to create novel 3D shapes and textures. GET3D [61] trains a model to generate shape and texture via a DMTet [189] mesh extractor and 2D adversarial losses.

Shape Generation. A recent approach called DreamFusion [162] introduces the use of pre-trained diffusion models to generate 3D NeRF [147] models based on a given text prompt. The key component in DreamFusion is the score distillation sampling (SDS), which uses a pre-trained 2D diffusion model as a critique to minimize the distribution of the predicted and ground-truth Gaussian noise, thus the 3D scene can be optimized for desired shape and texture generation. An recent method, named Fantaisia3D [34], disentangles the 3D human generation into shape and texture generation processes, which can generate high-quality 3D meshes by using normal maps. As depicted in Figure 4.1, we follow the design of Fantasia3D

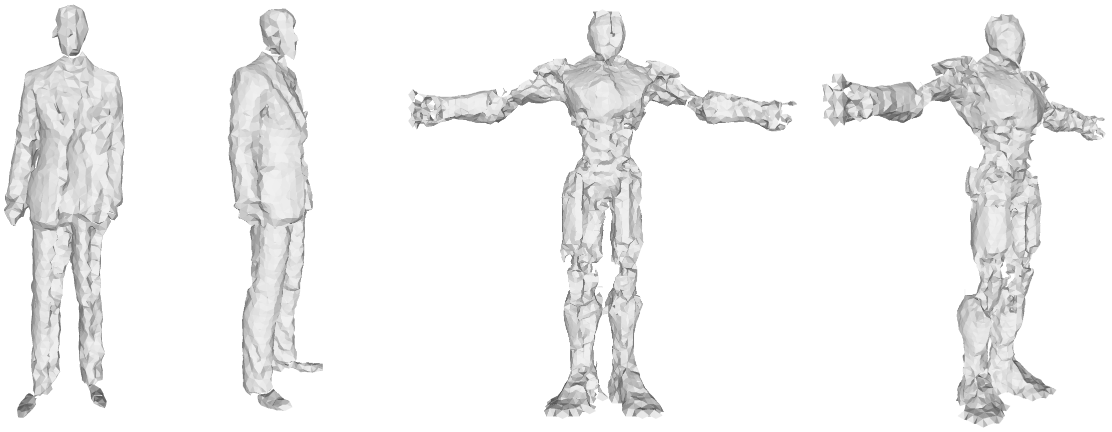


FIGURE 4.1. Generated results of human avatars guided by textural descriptions using modified Fantasia3D [34], where human prior information (i.e., SMPL) is injected during the training pipeline. Input texts are shown below the rendered images.

to generated human mesh models conditioned on input texts. Although the shape can be deformed to a large extent, the generated human shapes cannot remain consistent for the disjoint parts. The observed results indicate that current technologies for text-to-human shape generation is still far from satisfaction and the generated shapes are not semantically aligned to the input textual descriptions.

Texture Generation. In the context of texture generation, Latent-NeRF [144] demonstrated how the same SDS loss can be employed in the latent space of the diffusion model to generate textures for 3D meshes and then decoded to RGB for the final colorization output. Besides, both TEXTure [175] and Text2Tex [31] proposed a non-optimization method with progressive updates from multiple viewpoints to in-paint the texture over the 3D mesh models.

Human-specific shape and texture generation methods also follow the same ideas that use either CLIP similarity between the generated human image and the textural descriptions [83] or directly leverage SDS for iterative shape and texture generation [107, 258, 91]. Besides, they also employ human body model prior, i.e., SMPL [137], for effective human avatar generation. However, most generated human textures are over-smooth and of low quality, which we argue is caused by the unstable guidance provided by SDS. In this work, we present

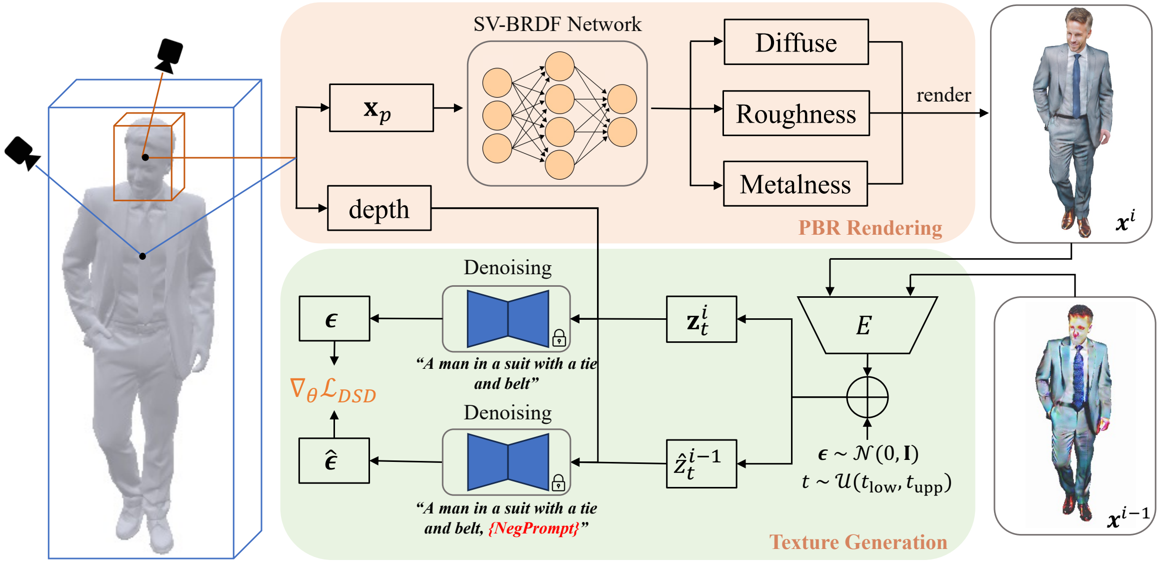

SV-BRDF Network
$\mathbf{x}_p$
depth
Diffuse
Roughness
Metalness
render
PBR Rendering
x^i
Denoising
ϵ
"A man in a suit with a tie and belt"
$\nabla_\theta \mathcal{L}_{DSD}$
Denoising
$\hat{\epsilon}$
"A man in a suit with a tie and belt, {NegPrompt}"
$\mathbf{z}_t^i$
$\hat{z}_t^{i-1}$
E
$\epsilon \sim \mathcal{N}(0, \mathbf{I})$
$t \sim \mathcal{U}(t_{\text{low}}, t_{\text{upp}})$
Texture Generation
x^{i-1}

FIGURE 4.2. Overview of our proposed model for text-to-human texture generation. Given an input text and a human mesh model, we generate a avatar texture to match the textual description. To achieve this, we propose Denoised Score Distillation with a negative pair of image and text prompts to guide the gradient direction for detail texture generation that is semantically aligned to the input text. We utilize a coordinate-based network with SV-BRDF to learn the material-related parameters (i.e., $\mathbf{k}_d$, r, and m) with depth maps for geometry-aware texturing. Finally, camera position is shifted with semantically adapted text input to refine the face region.

an approach that utilizes a modified SDS for the generation of high-quality, detailed textures while incorporating geometry-aware texturing techniques for intricate garment detailing.

4.3 Methods

4.3.1 Denoised Score Distillation

Given a textureless human avatar, our task is to generate surface textures conditioned on input texts. Due to SDS and neural representation of 3D avatar [147], zero-shot human texture generation is made possible. We observe that using SDS only for human texturing can cause over-smoothed body parts and cannot be fully semantically aligned to the input text.

We address the issue brought by SDS by proposing a new method, Denoised Score Distillation, for detailed human avatar texturing of high quality. Specifically, when presented with input text embedding y and the corresponding image $\mathbf{x}$ with the latent code $\mathbf{z}$, we aim to refine the gradient $\nabla_\theta \mathcal{L}_{\text{SDS}}$ in Eq. 4.4 to a direction, so that the rendered avatar contains a detailed texture mapping that is semantically aligned to the input text. Mathematically, our DSD score function is formulated as:

$$\mathcal{L}_{\text{DSD}} = w(t)\big(\|\epsilon_\phi(\mathbf{z}_t^i, y, t) - \epsilon\|_2^2 - \lambda\|\epsilon_\phi(\hat{\mathbf{z}}_t^{i-1}, \hat{y}, t) - \epsilon\|_2^2\big), \tag{4.8}$$

where we introduce a *negative* pair of image with latent code $\hat{\mathbf{z}}$ and text with embedding $\hat{y}$. λ is a weighting parameter. Both $\mathbf{z}_t^i$ and $\hat{\mathbf{z}}_t^{i-1}$ have a superscript i indicating the training iteration and share the same timestep t and noise ϵ, allowing us to use the same U-Net for noise prediction. Then the gradient of $\mathcal{L}_{\text{DSD}}$ over the model parameter θ is:

$$\begin{aligned} \nabla_\theta \mathcal{L}_{\text{DSD}} &= w(t)\big(\hat{\epsilon}_\phi\left(\mathbf{z}_t, y, t\right) - \epsilon - \lambda(\hat{\epsilon}_\phi\left(\hat{\mathbf{z}}_t, \hat{y}, t\right) - \epsilon)\big)\frac{\partial \mathbf{z}_t}{\partial \theta} \\ &= w(t)\big(\hat{\epsilon}_\phi\left(\mathbf{z}_t, y, t\right) - \lambda\hat{\epsilon}_\phi\left(\hat{\mathbf{z}}_t, \hat{y}, t\right) - (1-\lambda)\epsilon\big)\frac{\partial \mathbf{z}_t}{\partial \theta}, \end{aligned} \tag{4.9}$$

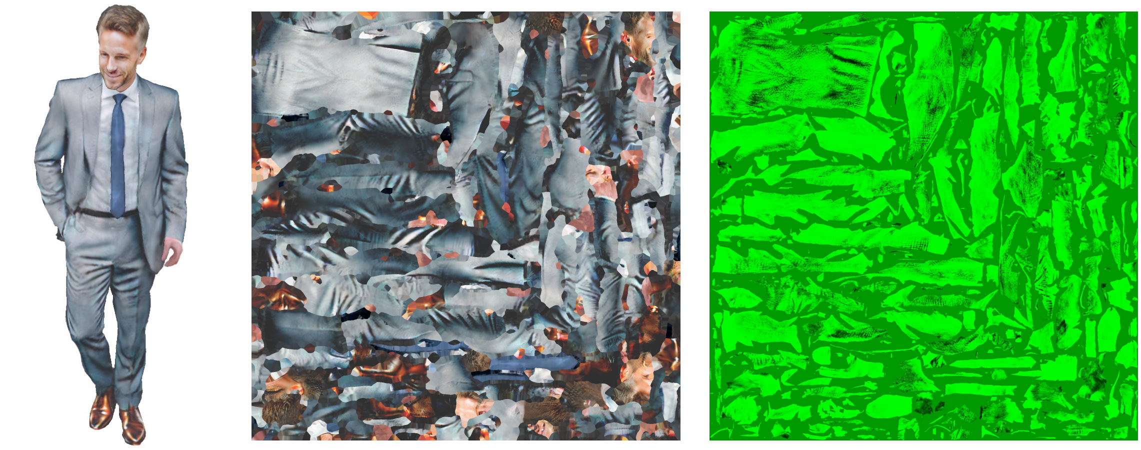
where we have omitted the U-Net Jacobian matrix following [162].

As depicted in Figure 4.2, we employ the negative image $\hat{\mathbf{x}}^{i-1}$ derived from the preceding training iteration, where we consider $\hat{\mathbf{x}}^{i-1}$ a negative version of $\mathbf{x}^i$ as it contains more noise signals. The inclusion of the negative image within the computation process of $\nabla_\theta \mathcal{L}_{\text{DSD}}$ yields two significant advantages. Firstly, $\hat{\mathbf{z}}_t^{i-1}$ can reinforce the memory of the rendered human image during long time training, so that the final output can still be semantically aligned to the input text. Secondly, the incorporation of the negative image improves the model's capacity to learn complex geometries, thus facilitating the generation of clear boundaries between varying garment types. For negative prompts, we use the common prompts such as *disfigured*, *ugly*, etc. However, we would adapt existing prompts based on a test run, infusing refined negative prompts based on the observed output. For instance, if artifacts emerge within rendered hand regions, we append "*bad hands*" to the prompt set. In contrast to the indirect application of negative prompts in Stable Diffusion, we inject the negative prompt embedding directly into $\nabla_\theta \mathcal{L}_{\text{DSD}}$. This strategy effectively minimizes artifact presence in the rendered human images, thereby enhancing the quality of the generated output.

Through the integration of both negative image and prompts, we successfully manipulate the existing SDS gradient in Eq. 4.4 to guide the model convergence towards a mode that yields highly detailed and qualitative textures, which also remain semantically aligned to the input text. Further analyses and insights into this approach are provided in our ablation study.

4.3.2 Geometry-aware Texture Generation

To accurately texture the details proposed by complex garments, we leverage depth map as a fine-grained guidance. Therefore, we employ a pre-trained depth-to-image diffusion model [176] rather than the general version, so that the generated avatar could follow the same depth values of the given surface mesh. Based on the depth-aware diffusion model, we find that the direct application of SDS tends to guide the model towards a specific mode [224], resulting in over-smoothed and noisy garment texture mappings. In addition, the generated texture is not semantically aligned to input prompts as the belt texture is not clearly presented in the rendered image. We also employ the differentiable rendering process and coordinated-based neural networks for more detailed geometry learning during the texture generation process. In addition, as depicted in Figure 4.3, we apply texture dilation on UV islands for to reduce texture seams for more smooth rendering.



(a) Rendered Image

(b) Texture map k_d

(c) Texture map k_s

FIGURE 4.3. Generated texture maps for the diffuse term $\mathbf{k}_d$ and specular term $\mathbf{k}_s$. Texture dilation on UV islands is only applied on $\mathbf{k}_d$ to reduce texture seams.

4.3.3 Semantic Zoom

Human perception is particularly sensitive to distortions and artifacts in facial features. However, texturing human avatars in a full-body context often results in degraded facial details. To address this issue, we enhance the human prior during the optimization process by semantically augmenting the prompt [83]. For instance, we pre-pend “the face of” to the beginning of the prompt to direct more attention to this region. Simultaneously, every four iterations, we shift the look-at point of the camera to the face center and semantically zoom into the facial region, which refines facial features and improves the overall perception of the rendered avatar.

4.4 Experiments and Results

4.4.1 Datasets

In this chapter, we utilize the RenderPeople dataset [174] as the input, where we intentionally dismiss the original textures given in the dataset. RenderPeople includes real-world scanned human data with both high- and low-resolution mesh options. For texturing tasks, we employ the low-resolution models where the number of vertices is only 30k for an efficient model training.

4.4.2 Evaluation Metrics

We apply both user study and CLIP score for the evaluation of the generated texture quality. Specifically, for user study, we pick several rendered images and the corresponding texts for evaluation. Totally, 345 users are picked with responses received. Each user is asked with two questions: (Q1) How closely does the result match the text description; (Q2) How realistic is the generated result. We ask users to rank their score from 1 to 5, where 1 indicates the most dissatisfied result and 5 indicates the most satisfied result. For CLIP score, we follow the same implementation of [168].

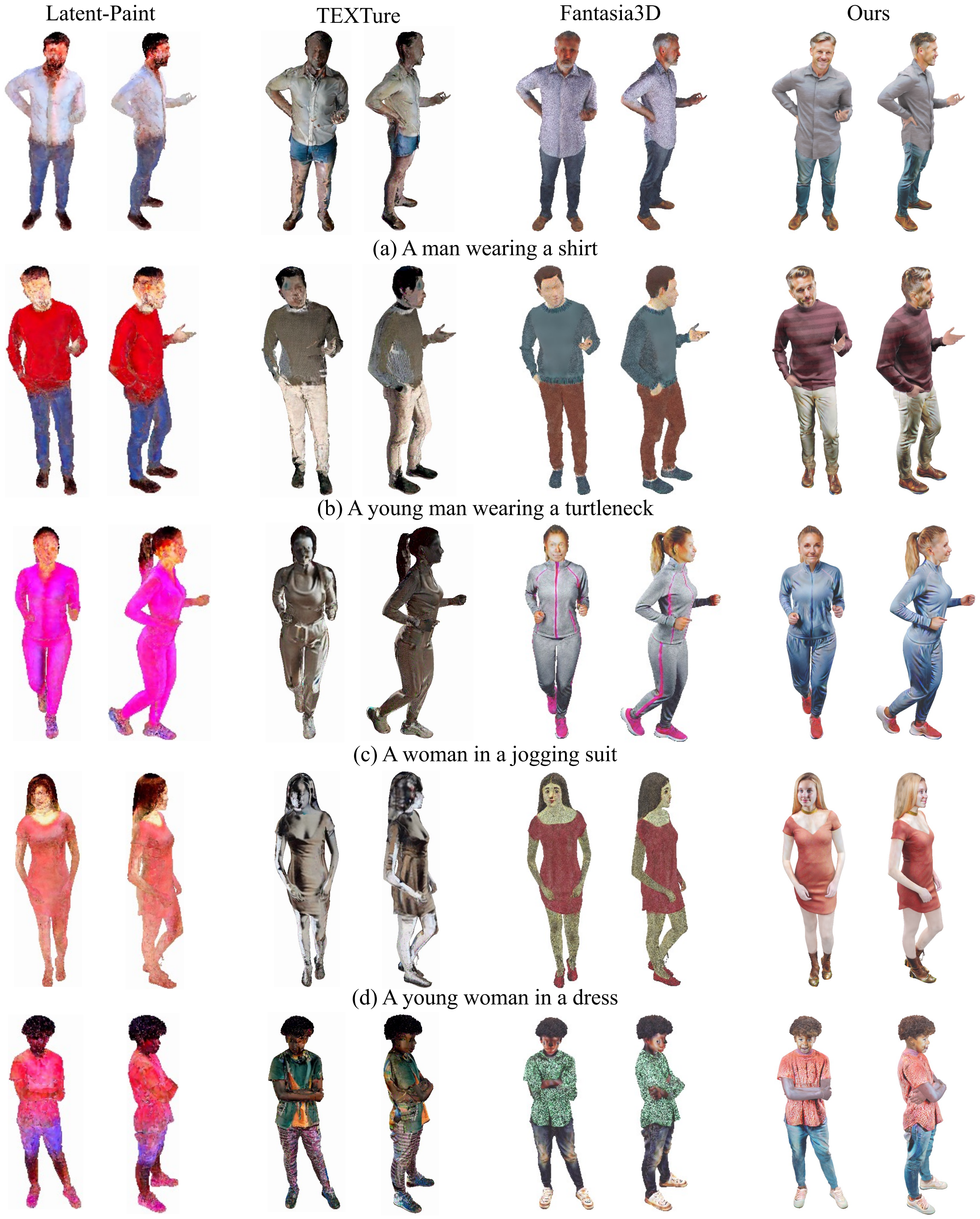
4.4.3 Implementation Details

We implement the SV-BRDF network in Figure 4.2 as a two-layer MLP with 32 hidden units. We train and optimize our method on one Nvidia RTX 3090 GPU for 6000 iterations. We use the AdamW optimizer with a learning rate of 1×10^{-3} for texture generation. We employ semantic zoom and shift our camera position to the face regions every 4 training iterations.

We compare our model to recent state-of-the-art baseline models, including Latent-Paint [144], TEXTure [175], and Fantasia3D [34] with the appearance modeling part only. We modify Fantasia3D to ensure the vertex positions remain fixed whiling generating textures. We also compare our model performance with a recent method for realistic human avatar generation, DreamHuman [107], to further validate the effectiveness of our design. Although the human mesh model is not publicly available, we use the same text prompts as in DreamHuman to evaluate the quality of human textures with similar human mesh models.

4.4.4 Qualitative Results

As depicted in Figure 4.4, we compare our qualitative results against baseline models. Latent-Paint is unable to capture the semantics of the objects, which results in failed or blurry textured avatars. TEXTure generates relatively better results than Latent-Paint, while it still suffers from inconsistent textures. Fantasia3D performs well given certain input texts as in Figure 4.4 (a) and (c). By using SDS which causes the unstable loss gradient direction, Fantasia produces unrealistic samples with noisy textures in most cases. In contrast, our model can output realistic textured avatars with high-quality and detailed textures, which are aligned to input texts and consistent with the geometry. We further compare our results with DreamHuman in Figure 4.5. We observe that using the same text input, our model generates textured avatars with more high-frequency details, such as the cloth wrinkles, which is different from DreamHuman where the textures are over-smoothed. Moreover, in both experiments, our model can consistently generate high-quality human faces. We attribute our advantage over the aforementioned baseline models to the proposed DSD score function,



(a) A man wearing a shirt

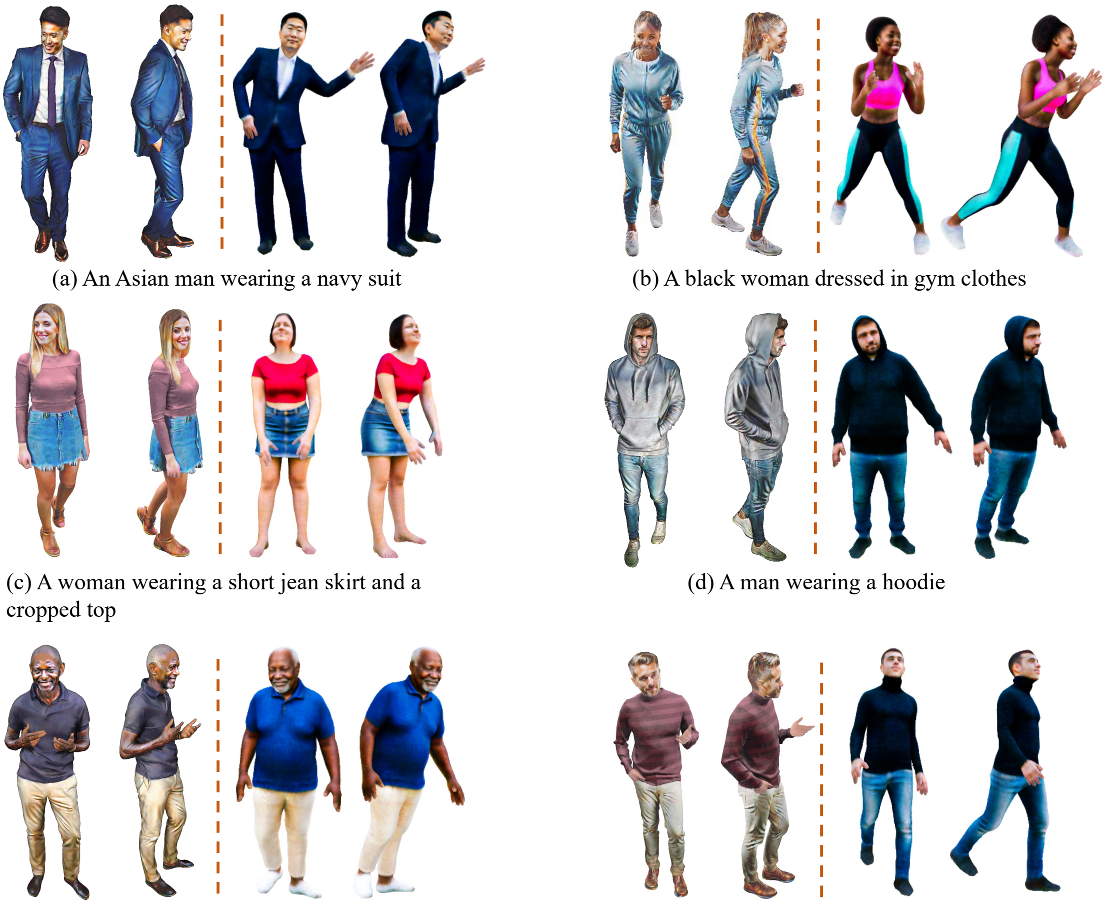
(b) A young man wearing a turtleneck

(c) A woman in a jogging suit

(d) A young woman in a dress

(e) A full-body shot of a boy with afro hair

FIGURE 4.4. Qualitative comparisons on RenderPeople [174] for textured human avatars against Latent-Paint [144], TEXTure [175], and Fantasia3D [34]. Our generation contains the best texture quality with high-frequency details and consistent with input textual descriptions.



(a) An Asian man wearing a navy suit

(b) A black woman dressed in gym clothes

(c) A woman wearing a short jean skirt and a cropped top

(d) A man wearing a hoodie

(e) A senior black person wearing a polo shirt

(f) A young man wearing a turtleneck

FIGURE 4.5. Qualitative comparisons with DreamHuman [107]. As DreamHuman is not publicly available, we pick similar mesh models from RenderPeople [174] and download the results from the published paper.

which guides the gradient in a direction that mitigates the over-smoothing artifacts commonly introduced by SDS.

4.4.5 Quantitative Results

To investigate the alignment between the rendered human avatars and the input texts, we leverage the CLIP score [168]. As shown in Table 4.1, we compare our method against the baseline models and report the mean CLIP score. Specifically, we generate six frontal images from all textured avatars, each separated by a 30-degree interval. we observe that our model outperforms all baseline models, where our result is higher than Latent-Paint by the largest

TABLE 4.1. Quantitative comparisons of mean CLIP score between baseline models and ours for textured human avatars. Δ denotes the percentage by which our model outperforms the indicated method.

Method	Mean CLIP Score	Δ (%)
Latent-Paint	24.11	19.99
TEXTure	25.34	14.17
Fantasia3D	27.10	6.75
Ours	28.93	-
DreamHuman	25.79	12.25
Ours	28.95	-

TABLE 4.2. User study results of baseline models and ours. Δ denotes the percentage by which our model outperforms the indicated method.

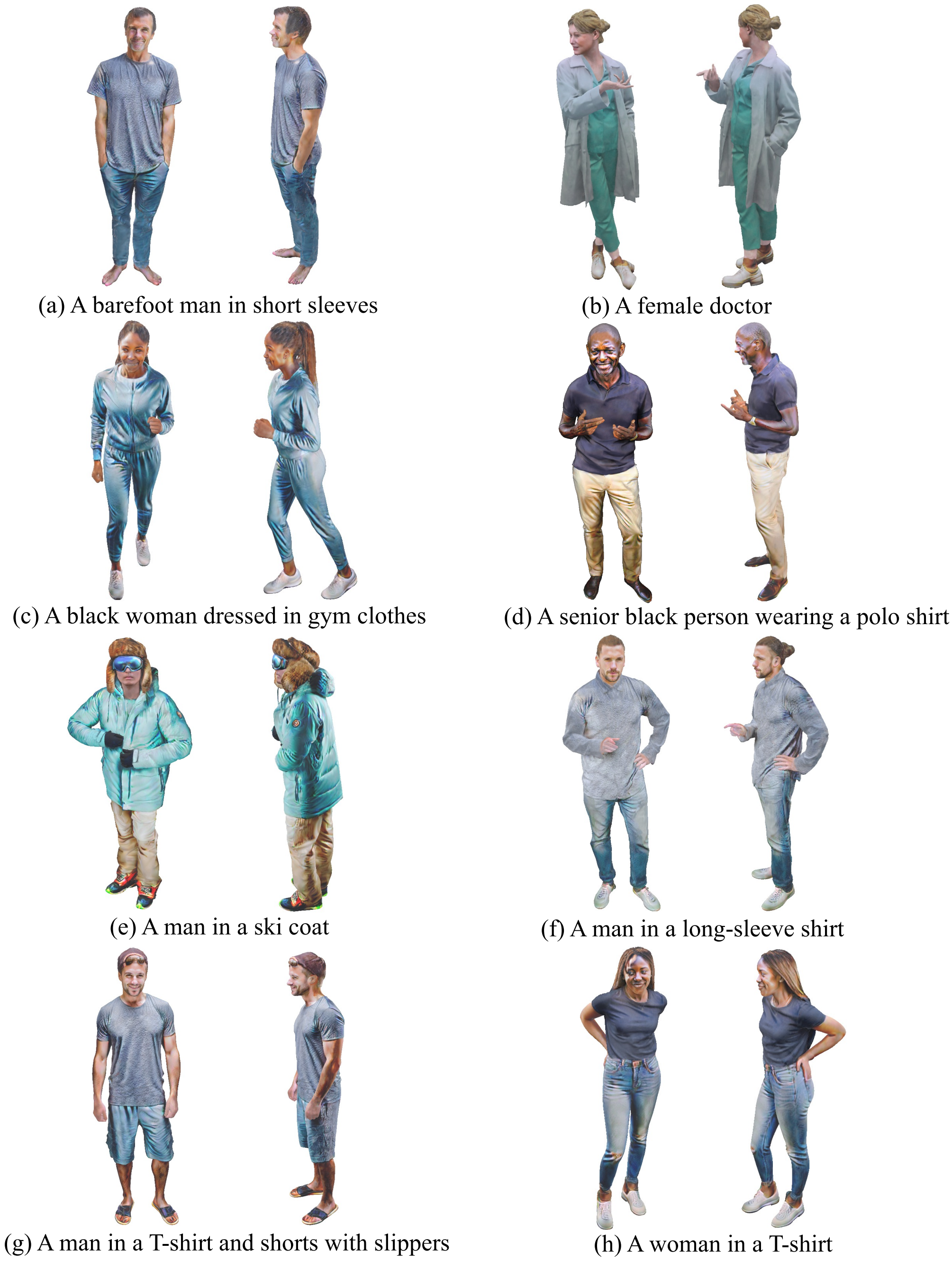
Method	Score	Δ (%)
Latent-Paint [144]	1.21±0.46	180.99%
TEXTure [175]	1.46±0.54	132.88%
Fantasia3D [34]	1.94±0.78	75.26%
DreamHuman [107]	3.00±0.80	13.33%
PaintHuman (Ours)	**3.40±1.09**	-

margin of around 19.99%. Such improvements demonstrate that our proposed DSD is capable of generating more realistic textures on complex human meshes, and is better aligned to the input texts.

Moreover, we conduct user studies for more accurate quantitative analysis. Collected results are reported in Table 4.2 including mean scores and standard deviations, which indicate that our PaintHuman model outperforms the other baselines. More qualitative results are shown in Figure 4.6.

4.4.6 Ablation Study

To validate the effectiveness of our proposed components, we use “a man in a suit with a belt and tie" as an example text prompt for the ablation study. Results of alternative settings are shown in Figures 4.7, 4.8, and 4.9.



(a) A barefoot man in short sleeves

(b) A female doctor

(c) A black woman dressed in gym clothes

(d) A senior black person wearing a polo shirt

(e) A man in a ski coat

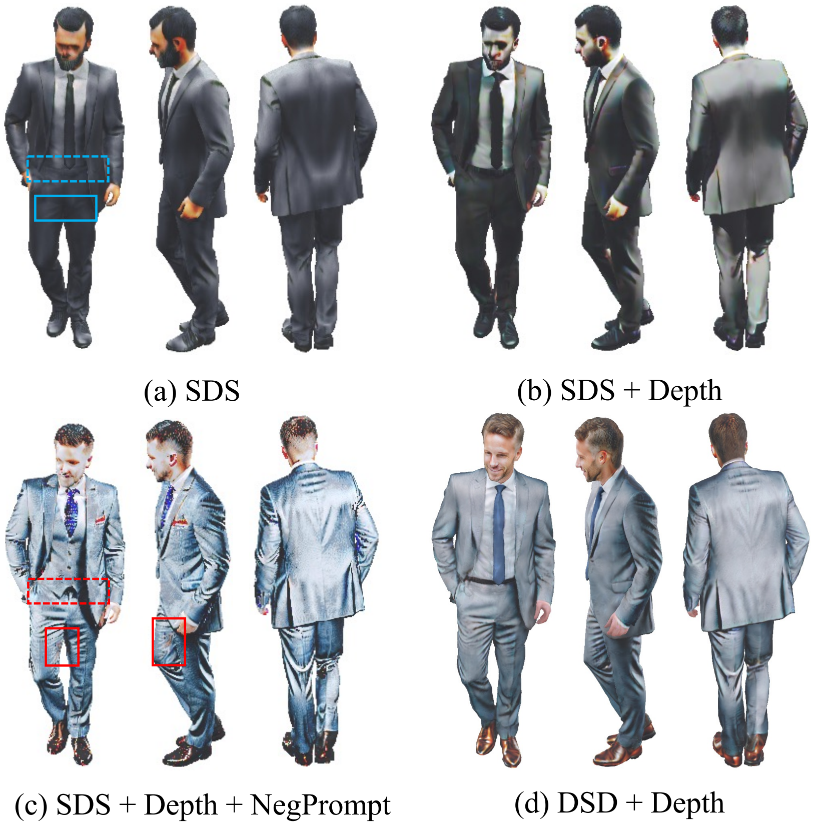
(f) A man in a long-sleeve shirt

(g) A man in a T-shirt and shorts with slippers

(h) A woman in a T-shirt

FIGURE 4.6. Additional rendered results on RenderPeople [174] for textured human avatars guided by textual descriptions, where the corresponding input text is shown at the bottom of each sample.

Firstly, the efficacy of our DSD is verified through several comparisons. As shown in Figure 4.7(a), we note that employing SDS for human texturing often results in over-smoothed body parts and fails to fully align with the input text semantically, where the belt region is



(a) SDS

(b) SDS + Depth

(c) SDS + Depth + NegPrompt

(d) DSD + Depth

FIGURE 4.7. Rendered results of textured Human avatars based on (a) SDS, (b) SDS with the depth map, (c) SDS with the depth map and negative prompts, and (d) DSD with the depth map. The dashed boxes indicate the mismatch between the texture and the textual descriptions, and the solid boxes denote human parts with low-quality textures.

neglected. The addition of depth map guidance in Figure 4.7 also struggles to address this issue. Moreover, by adding negative prompts, Figure 4.7(c) demonstrates that the rendered image is able to include more high-frequency details, but is not aligned with the input text, and some parts are devoid of texturing. In contrast, as shown in Figure 4.7(d), an image rendered using our DSD effectively mitigates the over-smoothing issue and results in a high-quality, detailed human avatar.

We further examine the effectiveness of BRDF shading model. As shown in Figure 4.8 (b), we render the result with the Spherical Harmonic model (SH) [21], resulting in less realistic textures with noticeably noisy color distributions at the borders between different garments. However, using BRDF can give us smooth and clear textures.

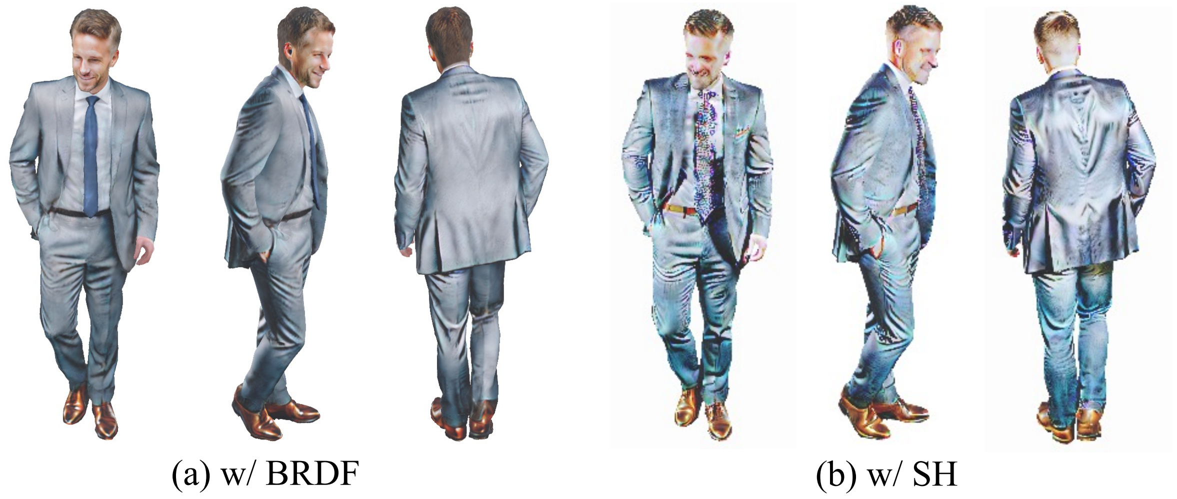


FIGURE 4.8. Ablation study on different shading models. (a) Fully-proposed method with BRDF; (b) Fully-proposed method with Spherical Harmonic model (SH).

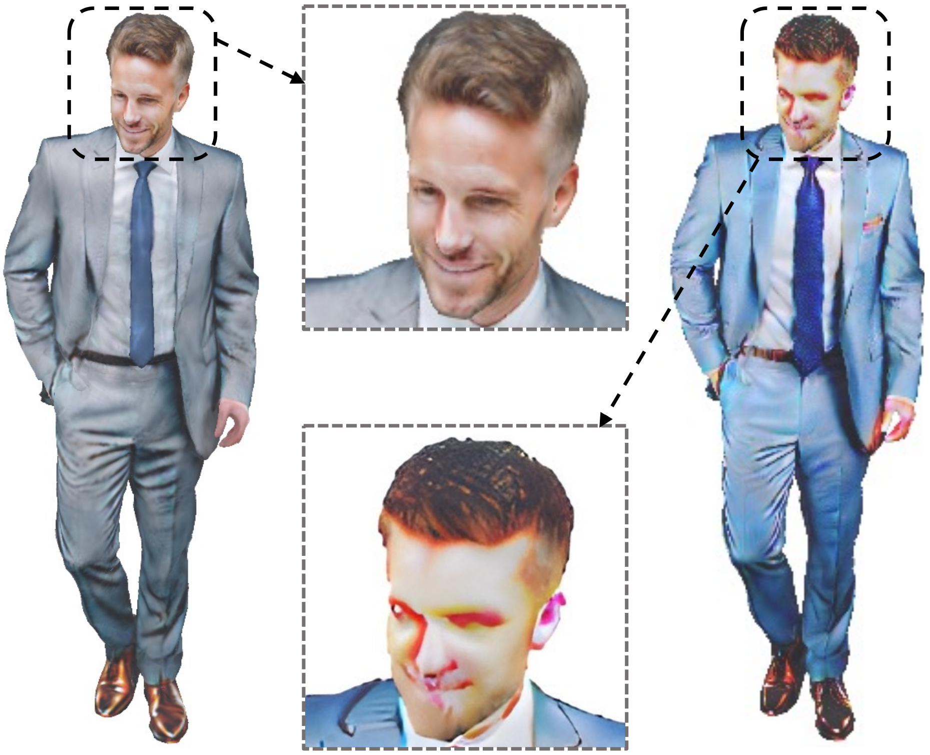
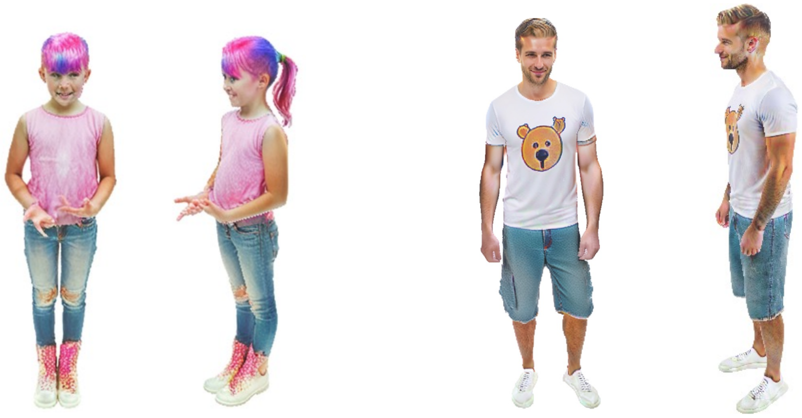


FIGURE 4.9. Importance of semantic zoom. The left image shows the generated avatar with semantic zoom, while the right image employs no semantic zoom.

Finally, as shown in Figure 4.9, our usage of semantic zoom on the face region significantly enhances the overall texture quality. Notably, the method enables the presence of intricate facial features, contributing to a more realistic representation.



(a) a girl with pink hair, bursting with vivid color

(b) a man wearing a T-shirt with a cute bear drawn on it

FIGURE 4.10. Texture generation with challenging text prompts.

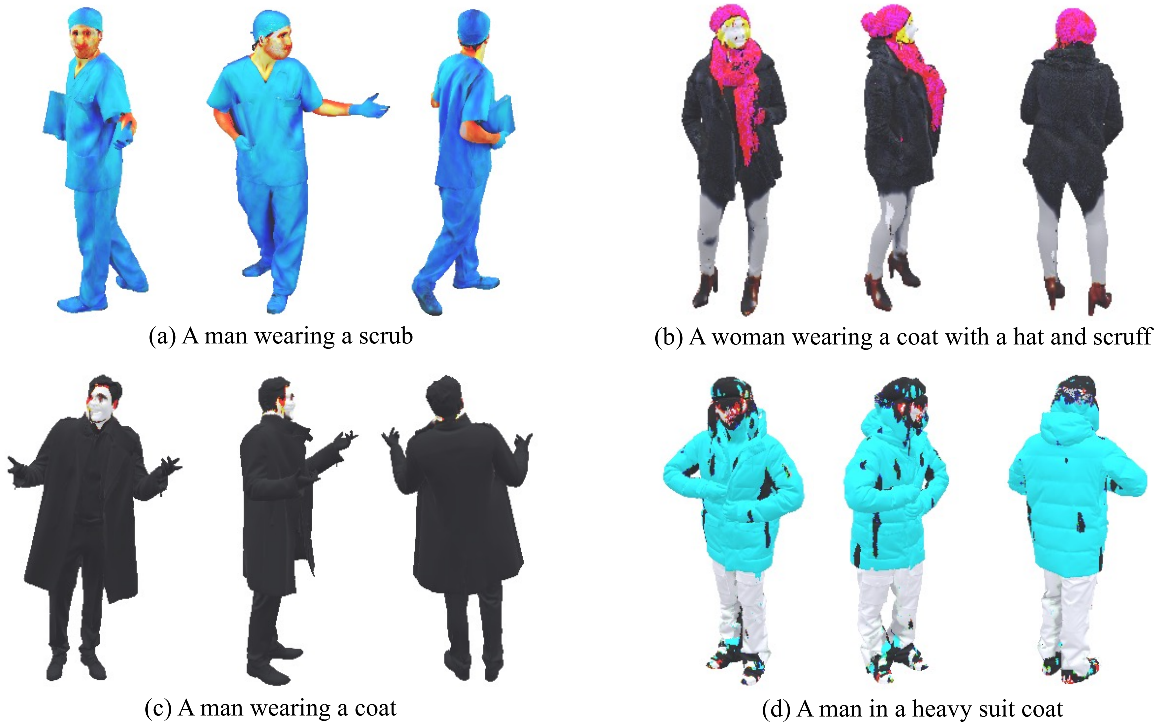
4.5 Discussion

Our proposed DSD can guide the texture generation process to a specific mode where the synthesized textures are semantically aligned to both input texts and complex geometry, and the textures are detailed.

In terms of practical use in industry, our design can quickly generate 3D models based on input texts with time less than 1 hour, which largely reduce the time expense. In addition, our output formats follow the standard of the industry and can be edited by the designers to fix some artifacts or to fit their requirements. Our method could offer an unprecedented level of personalization, allowing users to create unique avatars or objects simply by describing them in text.

However, our model currently cannot reflect fine-grained text prompts quite well. For example, as shown in Figure 4.10, when inputting texts that describe fine-grained properties, our model cannot render 3D models that reflect the prompts 100%. The limitation is caused by the ability of the language model that we use in our design, which could be improved by using a larger language model.

Moreover, we present our failed cases in Figure 4.11 to show that the proposed DSD is not able to handle all complex cases. As can be seen from Figure 4.11 (c), when the given human avatar is not in a canonical pose and when the hand gestures are quite complex, our model fails to paint the hand region, which also leads to the failure of the face region. Moreover, as shown in Figure 4.11 (d), our model can sometimes generate bad results on objects which can reflect light. For example, the snowboarding goggles are not accurately textured. Future works are essential to improve the current model performance.



(a) A man wearing a scrub

(b) A woman wearing a coat with a hat and scruff

(c) A man wearing a coat

(d) A man in a heavy suit coat

FIGURE 4.11. failure cases when using DSD for texture generation. Input textual descriptions for each sample are shown below the rendered images, where we show the front, side, and back views of the human avatar renderings.

4.6 Summary

In this chapter, we propose a zero-shot text-to-human texture generation model. We present Denoised Score Distillation, a novel method that refines gradient direction and produces high-quality and detailed human textures aligned to the input text. To ensure the semantic alignment between the mesh and texture, we leverage a pre-trained depth-to-image diffusion model and a coordinate-based network for surface material prediction, with a spatially-varying bidirectional reflectance distribution function for photorealistic human texturing. We also enhance facial details through semantic zooming. Extensive experiments demonstrated the effectiveness of our designs.

CHAPTER 5

Deep Learning-based Analysis for Micro-expression Face Videos

Face videos, as another type of 3D data, form a significant domain in 3D data analysis and processing and has far-reaching implications in numerous areas such as facial recognition, emotion detection, generation. With the rapid development of deep learning technologies, face video analysis is becoming increasingly critical, owing to the temporal and spatial information it encompasses that single image data fails to capture.

While the potential of face video data is evident, its analysis presents unique challenges. As the challenges mentioned in Section 1.3.1, we aim to address the dataset issue by firstly proposing a generative model for face videos with micro-expressions, where we have utilized and proposed a model based on a generative adversarial network (GAN) [63] in combination with a Capsule network for two subtasks: fake data discrimination and micro-expression recognition. Experimental results indicate that our GAN-based model can generate face videos that assists in recognizing facial micro-expressions

In this chapter, we explore the importance of the face video generation in the context of 3D data analysis, and how novel deep learning-based methodologies can be employed to harness this type of data more effectively.

5.1 Introduction

Video-based micro-expressions (MEs) display unconscious feelings that can be hardly perceived by untrained observers, making it a challenging recognition task. Due to the practical application of video-based micro-expression recognition (MER) in the domain such as lie detection and disease diagnosis [156, 143, 235], analysing face video data containing micro-expressions is quite important. However, since micro-expression face data is hard to collect, deep learning-based methods for MER cannot achieve a satisfactory result. Therefore, to address this issue, we introduce a module for micro-expression synthesis (MES) to increase the number of existing micro-expression face samples and introduce a simple classifier for MER task.

Before the application of deep learning algorithms, hand-engineered methods for recognizing MEs are used. For example, Facial Action Coding System (FACS) [44] is applied to recognize facial expressions, which pays attention to muscles that produce the expressions and measures the movement with the help of action units (AUs). Local binary pattern (LBP) and local quantized pattern (LQP) are later developed [143], and LBP with three orthogonal planes (LBP-TOP [265]) has shown superiority in processing facial images. However, these geometry-based methods rely heavily on the proposed images and can be easily affected by global changes.

With the development of deep learning technologies, lots of works have been proposed based on data-driven approaches for MER [101, 130, 209]. For example, ELRCN [101] adopts the model architecture of [54] with enriched features to capture subtle facial movements from frame sequences. Moreover, methods utilize the optical flow to enhance the model performance. STSTNet [123] extracts three optical features for lightweight network construction. Dual-Inception [268] learns the facial MEs with the help of horizontal and vertical optical flow components. ME-Recognizer [130] obtains optical features to encode the subtle face motion with domain adaptation, which achieves the 1^{st} place in MEGC2019 [186]. However, CNN-based methods are invariant to translations and do not encode positional relations, and orientation information of different facial entities is also ignored. Furthermore, due to the quantity limit of ME samples, the model performance is heavily constrained.

In addition, the size limitation of current ME datasets (i.e., SMIC [117], CASME II [246], and SAMM [48]) puts a constraint on deep learning methods that can hardly derive benefits from small scale datasets. We propose to increase the number of data samples by adopting GAN [63] and the variants [148, 157, 169], which have shown significant generative capabilities in

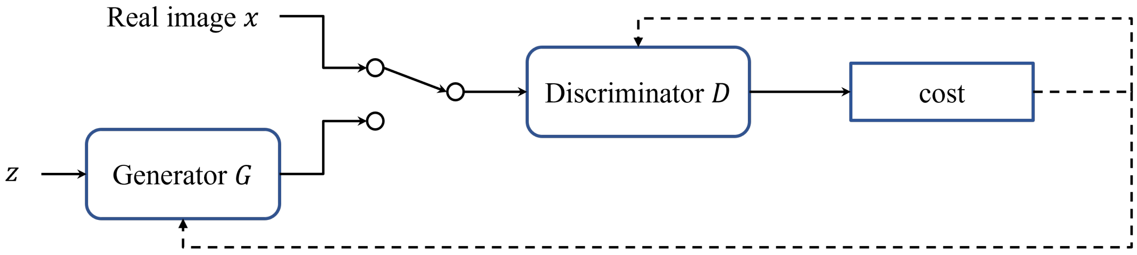


FIGURE 5.1. The model pipeline for GAN [63].

various vision application fields. Moreover, we utilize a graph convolution network [104] for facial parts relation learning, where we construct graphs on the feature dimension for dynamic feature relation learning.

The main contributions are summarized as 3-fold:

(1) We develop an Identity-aware and Capsule-Enhanced Generative Adversarial Network with graph-based reasoning, namely ICE-GAN, for MES and MER.

(2) We design a generation module that produces face videos with controllable MEs based on identity information and a graph reasoning module.

(3) We introduce a capsule-enhanced discriminator to distinguish image authenticity and predict micro-expression labels, with position-insensitive issues alleviated by the capsule-based algorithm to improve the MER accuracy.

5.2 Literature Review

5.2.1 Generative Adversarial Network

Introduced by Goodfellow [63], GANs are a type of unsupervised learning model that consist of two neural networks, the Generator and the Discriminator, working adversarially to outperform each other (see Figure 5.1). The generator network creates fake data samples, trying to produce data that come from the same distribution as the training set. Meanwhile, the discriminator network aims to distinguish between samples from the true data distribution (training set) and the face data created by the generator. Here, we give a review of GANs.

Give a latent variable z sampled from a normal distribution $\mathcal{N}(0, I)$, the generator G aims to map the latent variable to the data space, while the discriminator D outputs a single scalar representing the probability whether the input data comes from the real data or the generator. They are trained in a min-max game, where the generator tries to fool the discriminator, and the discriminator tries to correctly classify real from fake samples. The mathematical expressions for the generator and the discriminator are shown as follows:

$$\begin{aligned} \text{Discriminator: } \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right], \\ \text{Generator: } \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right), \end{aligned} \tag{5.1}$$

where θ_d is the learnable model parameters for the discriminator, θ_g the learnable model parameters for the generator, and i indicates the i-th sample among totally m samples.

Various improvements have been proposed to address the initial issues associated with GANs, such as mode collapse, vanishing gradients, and training instability. Radford et al. [169] proposes the Deep Convolutional GAN (DCGAN), which introduces architectural constraints that improving the training dynamics. Arjovsky et al. [9] present the Wasserstein GAN (WGAN) that utilizes the Earth Mover's distance, providing a more stable training process and preventing mode collapse. Later works develop Conditional GANs (CGANs) [148], where both the generator and discriminator are conditioned on some auxiliary information like class labels or data from other modalities, enabling the generation of data with specified characteristics.

Existing datasets for the micro-expression recognition task constrain the development of data-driven methods as the number of data samples is too small, resulting in overfitting during the inference stage. To enlarge the current dataset, GANs are applied to generate unseen videos by modeling the training data distribution, where distribution mappings are learned by the generator and the authenticity of inputs is determined by the discriminator. So far there have been tremendous extensions of GANs, and applications of GANs on the ME domain are also explored [124, 236]. For example, optical flow images are generated via GAN to enlarge the dataset in [124], and real facial images are produced based on action units intensity in

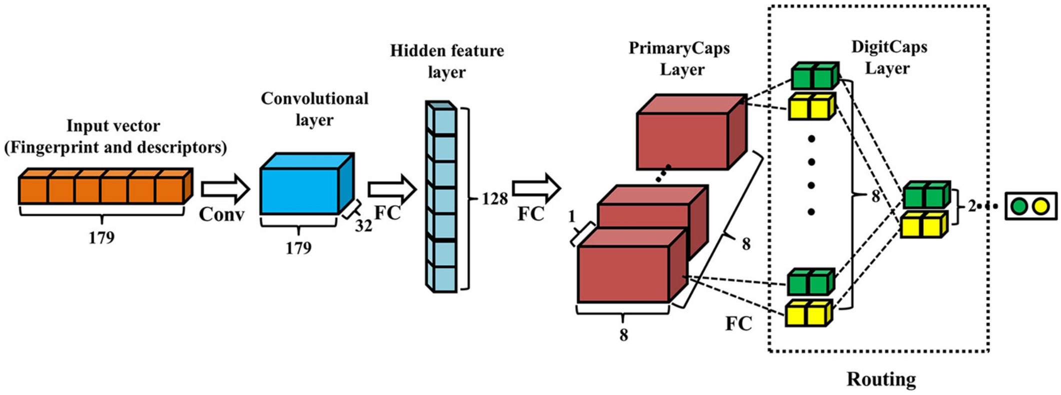


FIGURE 5.2. Detailed model architecture of the capsule network [220, 180].

[236]. Our method is inspired by Auxiliary Classifier GAN (ACGAN) [157] to generate unseen MEs based on image features with controllable categories, which can be of high quality and high discriminability.

5.2.2 Capsule Network

Different from conventional CNNs, capsule network contains capsules which consider the spatial hierarchy between features, ensuring that patterns are recognized in the context of their spatial relationship. The capsule network is translation equivariant, which presents a competitive learning capacity considering the relative pose and position information of object entities in the image. Moreover, instead of pooling layers used in CNNs, capsule network uses a dynamic routing mechanism to decide where to send the output. This ensures that the network considers the spatial relationships between features. Capsule network tends to be more robust against adversarial attacks compared to CNNs, and can achieve a better or comparable performance with fewer parameters than CNNs. The model design of the capsule network is depicted in Figure 5.2.

Ertugrul et al. [158] manages to encode face poses and AUs at different view angles with the help of a single capsule. LaLonde et al. [109] develop a deconvolutional capsule network with U-net architecture [177] which achieves a good result in object segmentation. [209] is the first work that manages to use capsule-based architecture for MER, whose model performance is better than the LBP-TOP baseline and several CNN models. In contrast, we propose our

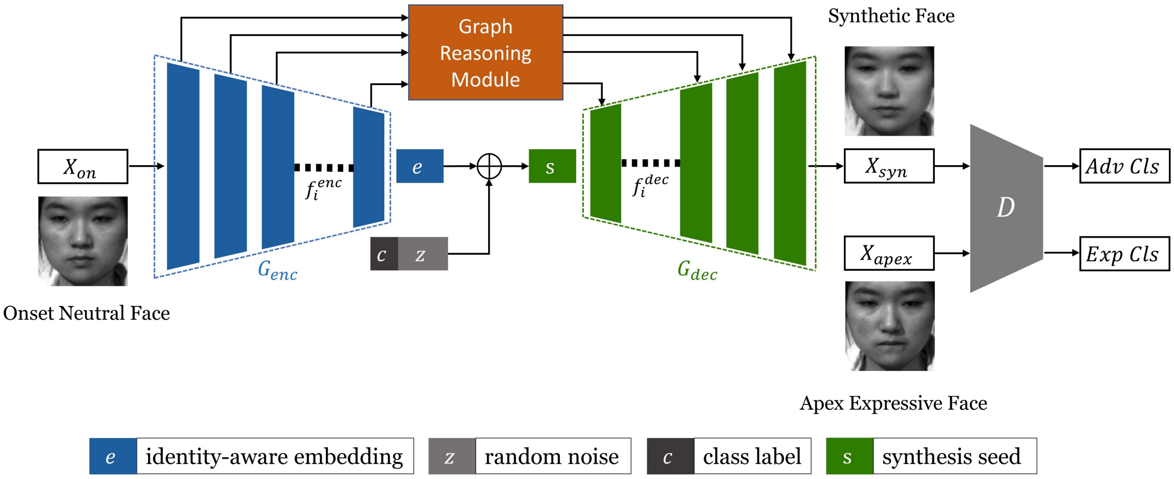


FIGURE 5.3. Architecture of the proposed ICE-GAN framework for MES and MER tasks, where G_{enc} and G_{dec} represent the encoder and decoder part of generator, and D denotes the capsule discriminator.

discriminator design which is enhanced by capsule network to implement multiple tasks: to check whether the input image is real or fake and to predict the ME labels.

5.3 Methods

The overall architecture is shown in Figure 5.3. The onset image X_{on} is regarded as the neutral face with the lowest expressive intensity. U-net like generator is used to modulate X_{on} with side information (i.e., random noise z and class label c) and produce the output X_{syn} with a desired class, preserving the identity knowledge. The real apex expressive faces X_{apex} and X_{syn} are adopted to train our capsule-enhanced discriminator, which is a multi-tasking component for authenticity checking between X_{apex} and X_{syn}, and for expression classification. More details are elaborated in the following sections.

5.3.1 Micro-Expression Synthesis via Identity-aware Generator

We propose an identity-aware generator G based on the encoder-decoder structure as shown in Figure 5.4, which can preserve the identity feature and produce outputs with preferable classes during MES, with the help of side information. The encoding procedure explores facial attributes from onset neutral inputs X_{on} via a series of hierarchical convolutional layers, resulting in intermediate feature maps $f^{enc}_{i:1,2,...,6} \in \mathcal{R}^{C_i \times H_i \times W_i}$ at the i-th convolutional layer, where C_i, H_i, and W_i denote the channel, height, and width of the feature map, respectively.

Furthermore, we decide to utilize f^{enc}_i by processing and propagating the intermediate information from the encoder G_{enc} to decoder G_{dec} for more realistic image generation, as these local features contain trivial knowledge such as 2D face geometries and non-trivial knowledge that is useful for the expression generation. So, we propose to use a reasoning module based on graphs, namely graph reasoning module (GRM), to capture facial part relations to reduce artifacts.

Instead of leveraging skip connections to directly transfer the multi-scale spatial information, we design to firstly flatten the spatial dimension and directly apply self-attention on feature channels to learn a channel map. Moreover, we propose to use graph convolution to better reason the relationships between feature channels by treating the channel map as a graph, which is named as the channel graph, so that global interdependencies between intermediate facial attributes can be captured from local feature responses.

The detailed architecture is displayed on the yellow box of Figure 5.4. Based on the feature map $f^{enc}_i \in \mathcal{R}^{C_i \times H_i \times W_i}$ after convolution, we can construct a spatial graph with the node number $N_s = H_i \times W_i$ and node feature $C_s = C_i$. Inspired by [229], super-nodes with richer expressiveness are formed by a transformation function $T(\cdot)$ to generate a new feature map $\hat{f}^{enc}_i \in \mathcal{R}^{C_i \times \hat{N}_s}$, and empirically, $T(\cdot)$ is chosen as a convolutional operation. To learn global relations between different facial encodings, self-attention is applied to calculate a similarity mapping of features based on $\hat{f}^{enc}_i$, which can be formulated as:

$$\mathcal{M} = \phi(\hat{f}^{enc}_i)\theta(\hat{f}^{enc}_i)^T, \tag{5.2}$$

where both $\phi(\cdot)$ and $\theta(\cdot)$ are linear mappings. We take the similarity mapping $\mathcal{M} \in \mathcal{R}^{C_i \times C_i}$ as a channel graph, which has a set of node features $\mathbf{m} = \{m_{j:1,2,...,C_i} \in \mathcal{R}^{1 \times C_i}\}$. In our case, we define $\mathcal{M}$ as an undirected and fully connected graph.

To reason over the whole channel graph, we utilize a GCN [104] to learn edge features and update node features. As the initial edge feature between two nodes is unknown and undefined, we propose to set up a self-learning adjacency matrix $A \in \mathcal{R}^{C_i \times C_i}$, which can be randomly initialized following the normal distribution and updated during back-propagation. The whole process can be expressed as:

$$\hat{\mathcal{M}} = \sigma((A + I)\mathcal{M}W), \tag{5.3}$$

where σ is an activation function, identity matrix I is added to construct self-looping, and $W \in \mathcal{R}^{C_i \times C_i}$ and $A \in \mathcal{R}^{C_i \times C_i}$ denote the weight and adjacency matrices, which are both learnable.

We then project the updated graph $\hat{\mathcal{M}}$ back to the same feature space of f_i^{enc} through an inverse projection function $T^{-1}(\cdot)$. Lastly, the residual connection is employed to compensate the high-level semantics relations with the low-level information. The final output g_i^{dec} of the graph reasoning module is represented as:

$$g_i^{dec} = f_i^{enc} \oplus T^{-1}(\hat{\mathcal{M}}). \tag{5.4}$$

The last layer of G_{enc} is the bottleneck layer that learns a compressed representation of the input data, termed as identity-aware embedding $e \in \mathcal{R}^{320 \times 1 \times 1}$. As a controlling term for expression generation, the class label c is concatenated with e and random noise z to form a synthesis seed $s \in \mathcal{R}^{423 \times 1 \times 1}$, which is fed into the decoding part where the spatial dimensions of feature maps are expanded and feature channels are reduced. Thus the decoding feature maps $f_{i:1,2,\ldots,6}^{dec}$ can be obtained with the help of deconvolution operation $h(\cdot)$ [136]. Moreover, g_i^{dec} will be further concatenated with f_i^{dec} so the learned relational information from latent feature space can be leveraged to improve synthetic image quality. The i-th upper layer feature map (for $i \in [2, 6]$) can be represented as:

$$f_{i-1}^{dec} = h(g_i^{dec} \oplus f_i^{dec}). \tag{5.5}$$

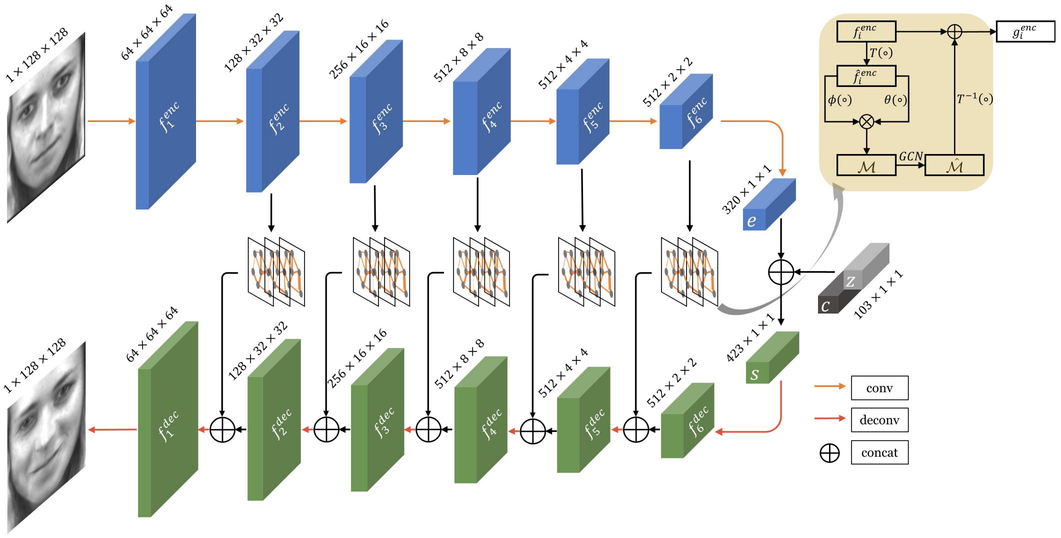


FIGURE 5.4. The encoder-decoder framework of our proposed generator G, where the detailed implementations of the graph construction and global reasoning are illustrated within the yellow box.

5.3.2 Micro-Expression Recognition via Capsule-Enhanced Discriminator

Unlike CNNs operating over single scalars, capsule network attends to vectors, of which lengths are used to represent the existence probabilities of each entity in a given image. We design a multi-tasking discriminator for sample authenticity checking and ME label classification enhanced by capsule network, which enables the learning of richer visual expressions and more sensitive to the geometric encoding of relative positions and poses of entities than conventional CNNs, dubbed a capsule-enhanced discriminator.

Figure 5.5 presents the detailed architecture. The facial attributes are encoded via *PatchGAN* for faster learning, which are then fed into *PrimaryCaps* to encapsulate the information at a lower level. Vectors generated from *PrimaryCaps* are coupled and used to activate the capsules in the next layer. Two following capsules, namely *AdvCaps* and *ExpCaps*, are designed for two separate sub-tasks: (1) *AdvCaps* distinguishes the real expressive images X_{apex} from the synthetic ones X_{syn}, and (2) *ExpCaps* predicts the corresponding micro-expression labels of input images. Furthermore, a reconstruction network [180] is desired for a performance gain by regularizing the training of *ExpCaps*.

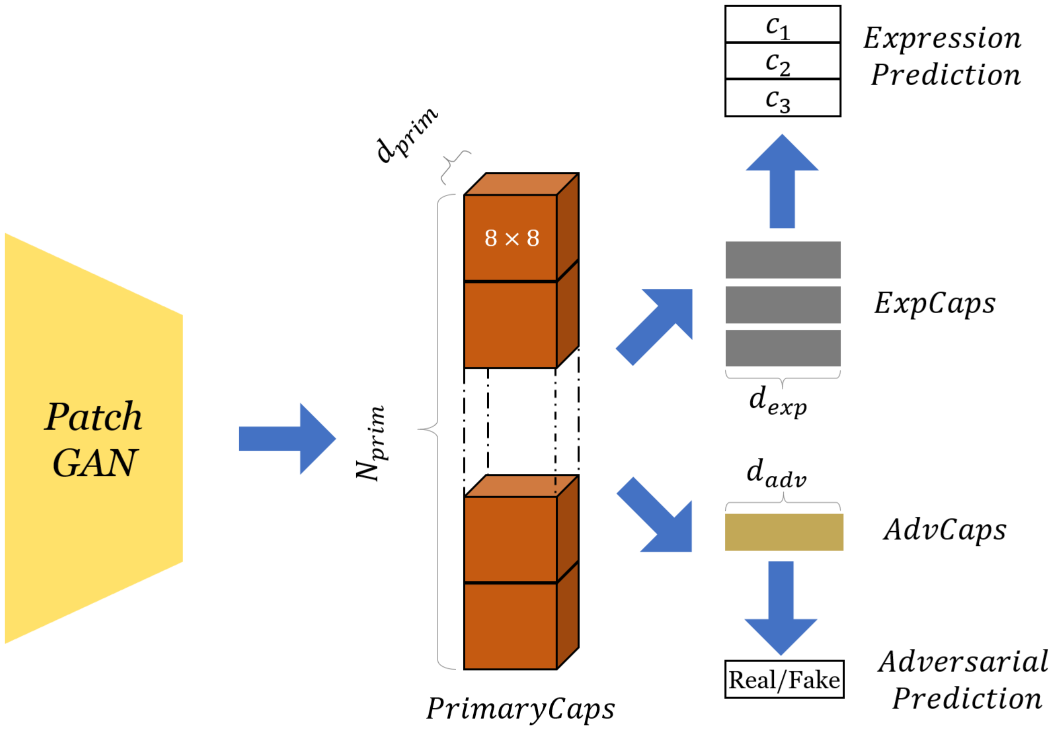


FIGURE 5.5. The capsule-enhanced discriminator D of our proposed method, where N_{prim} denotes the number of primary capsules, d_{prim}, d_{exp}, and d_{adv} represent the dimension of each *PrimaryCaps*, *ExpCaps*, and *AdvCaps*, respectively.

5.3.3 Model Objective

Generator Objective. A two-term identity-preserving loss L_{ip} is adopted to capture the identity information embedded in X_{on} (via G_{enc}) and thus to synthesize distinctive samples X_{syn} (via G_{dec}). One term of L_{ip} is selected as the pixel-wise reconstruction loss L_{pixel} to improve the generated image quality. We empirically use L1 penalty over L2 penalty to establish direct supervision on X_{syn}. Meanwhile, we preserve a perceptual similarity between X_{on} and X_{syn} during MES as in [94] by introducing a perceptual loss L_{per}, which is adopted as the second term in L_{ip} to preserve the facial styles with regards to different subjects. With the help of a cost network which is usually implemented as a pre-trained CNN, L_{per} can be easily estimated and minimized over high-level feature representations associated with X_{on} and X_{syn}. The overall representation of $L_{ip}(G)$ can be described as:

$$L_{ip}(G) = L_{pixel}(G) + \alpha L_{per}(G). \tag{5.6}$$

Discriminator Objective. The training of the capsule-enhanced discriminator is optimized by the margin loss L_{margin} as suggested in [180], which can enlarge the feature distance of different facial expressions. L_{margin} can be obtained via:

$$\begin{aligned} L_{margin}(D) &= T_k max(0, m^+ - \|\boldsymbol{v_k}\|)^2 + \\ &\lambda_k(1 - T_k)max(0, \|\boldsymbol{v_k}\| - m^-)^2, \end{aligned} \tag{5.7}$$

where T_k = 1 if expression class k exists otherwise 0, while m^+ and m^- are the upper and lower margins and v_k is the vector output of capsules that being activated to class k. Moreover, the mean square error is employed as the loss function L_{rec} in the reconstruction network to regularize the training procedure. Hence, the classification-related loss L_{cls} for our capsule-enhanced discriminator is summed as:

$$L_{cls}(D) = L_{margin}(D) + \beta L_{rec}(D). \tag{5.8}$$

Overall Objective. As we treat GAN as our baseline, the optimization process between the generator and discriminator is described as a min-max game [148]. The learning objective L_{gan} can be formulated as:

$$\begin{aligned} L_{gan} = \underset{G}{min}\,\underset{D}{max}\, V(D, G) &= E_{x \sim p_x}[log(D(x|y))] + \\ &E_{z \sim p_z}[1 - D(G(z|c))], \end{aligned} \tag{5.9}$$

where x and y indicate the real images and the real labels, while z and c denote the random noise and fake labels.

Overall, the total loss function of our proposed ICE-GAN can be summarized as:

$$L_{tot} = \lambda_{adv} L_{gan}(D, G) + \lambda_{mes} L_{ip}(G) + \lambda_{mer} L_{cls}(D), \tag{5.10}$$

where λ_{adv}, λ_{mes}, and λ_{mer} are multi-tasking weight parameters of the proposed tasks including GAN optimization, identity-aware MES, and capsule-enhanced MER.

5.4 Experiments and Results

5.4.1 Datasets

Evaluations are conducted on the MEGC2019 cross-database benchmark, which consists of three publicly available ME datasets: SMIC [117], CASME II [246], and SAMM [48].

SMIC: SMIC dataset consists of 164 micro-expression clips with 16 participants in the recording experiment, with a high speed camera of 100 fps used to record the short duration of micro-expressions. A sample from SMIC dataset is shown in Figure 5.6.

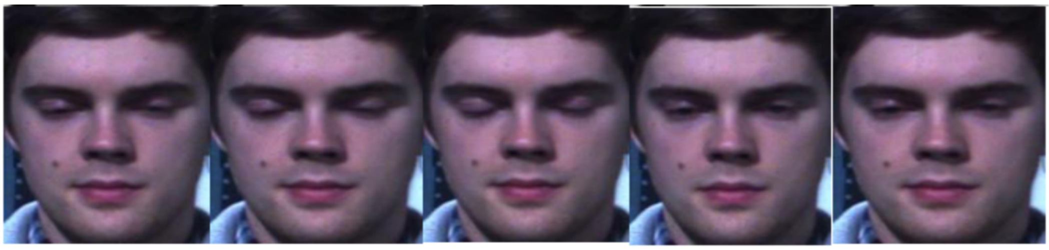


FIGURE 5.6. Extracted video samples from SMIC dataset [117].

CASME II: CASME II dataset consists of 247 micro-expression samples from 24 participants, and the samples are selected from nearly 3,000 elicited facial movements. CASME II has a sampling rate of 200 fps, which provides more detailed information on the facial muscle movements. In addition, the sample resolution in CASME II is 280×340. An extracted video sample from CASME II is shown in Figure 5.7.

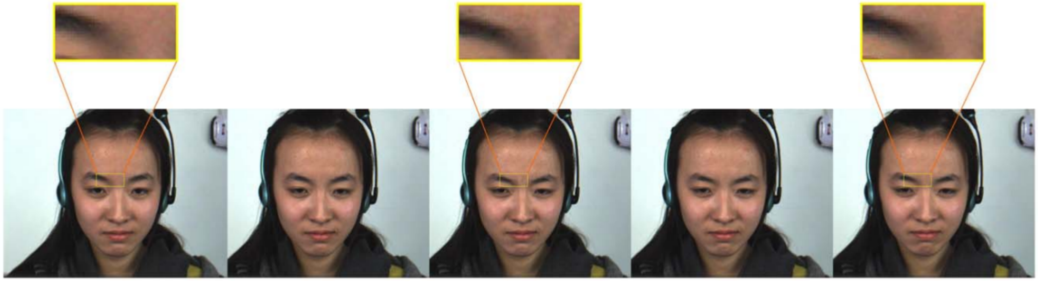


FIGURE 5.7. A demonstration of the frame sequence in a micro-expression from CASME II [246].

SAMM: SAMM dataset consists of 133 micro-expression video clips from 28 participants. SAMM has a sampling rate of 200 fps with the recorded face frame resolution set to 2040×1088, which has the highest frame resolution amongst all three datasets. A video sample from SAMM is shown in Figure 5.8.

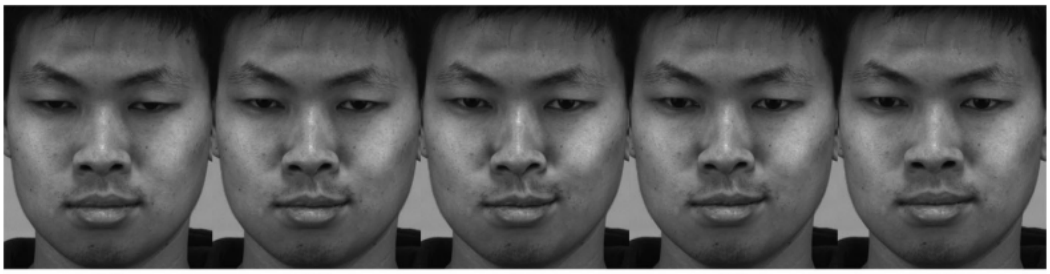


FIGURE 5.8. A video sequence from SAMM dataset 5.8.

5.4.2 Evaluation Metrics

In our case, frames in all 3 datasets are categorized into 3 classes of positive, negative, and surprise with the Leave-One-Subject-Out (LOSO) validation method. LOSO is conducted for subject-independent evaluation, i.e., the evaluation is repeated each time for all subjects accordingly, until each subject is split alone as the 1-subject testing dataset and the remaining subjects as the training dataset. We firstly conduct experiments based on cross-database evaluation (CDE), we follow the implementation in [186] by using the Unweighted F1-score (UF1) and Unweighted Average Recall (UAR) to make a fair comparison. We then evaluate the model performance only on CASME II and SAMM for single-database evaluation (SDE) using F1-score following [236].

5.4.3 Implementation Details

Data Preprocessing. We firstly cropped out face regions by conducting facial landmark detection repeatedly to locate and refine the positions of 68 facial landmarks [76], ending up with a bounding box decided by the refined 68 landmarks with toolkit [1]. We then followed

[1] https://pypi.org/project/face-recognition/

[209] to find the onset and apex frames in SMIC because annotated positions of onset and apex frames are only available in CASME II and SAMM. Moreover, we chose the neighboring four images around the apex image for each subject to augment the existing data, where we assumed that they all have the same expressive intensity. Finally, these cropped face images were resized to 128×128 in grayscale.

Model Implementation. The multi-tasking weight parameters in 5.10 are set as follows: λ_{adv} = 0.1, λ_{mes} = λ_{mer} = 1. Two reweighting ratios α and β are set to 0.1 and 5e-4. We set the noise dimension N_z to 100 and class label dimension N_c to 3 as we only have 3 classes. The discriminator includes a 70×70 *PatchGAN* followed by N_{prim} = 8 *PrimaryCaps* with dimension d_{prim} = 16 and two sub-capsules for classification purposes (d_{adv} = 256 and d_{exp} = 32). m^+, m^-, and λ_k in 5.7 are set to 0.9, 0.1, and 0.5. Adam is selected as the optimizer to train our network, with momentums b_1 and b_2 set to 0.9 and 0.999, respectively. The learning rate is initialized as 1e-3 and decays using a cosine annealing schedule. The batch size is set to 16 with 100 training epochs. The end-to-end training procedure of ICE-GAN was implemented in Pytorch with one Nvidia RTX2080Ti GPU.

5.4.4 Quantitative Analysis of MER

We compare the model performance against SoTA methods from MEGC2019 benchmark [186] using CDE. The results of UF1 and UAR are reported in Table 5.1. It can be observed that ICE-GAN outperforms approaches from MEGC2019 benchmark, where UF1 and UAR scores are improved by **10.9%** and **12.9%** compared to the best method [130].

The baseline (i.e., LBP-TOP) in Table 5.1 adopts hand-crafted features and receives a lower result compared to deep learning methods. CapsuleNet [209] utilizes capsule network to achieve an acceptable result, and the remaining works (i.e., [123, 268, 130]) propose their designs based on optical flows. The superior performance of our design can be attributed to the expanded size of the database and the global relations of facial attributes which are learned on channel graphs. Although ME-Recognizer [130] captures spatiotemporal information from

TABLE 5.1. MER results on the MEGC2019 benchmark and separate datasets in terms of UF1 and UAR with LOSO cross-database evaluation.

Method	MEGR 2019		SAMM		SMIC		CASME II	
	UF1	UAR	UF1	UAR	UF1	UAR	UF1	UAR
LBP-TOP [265]	0.588	0.578	0.395	0.410	0.200	0.528	0.702	0.742
CapsuleNet [209]	0.652	0.650	0.620	0.598	0.582	0.587	0.706	0.701
Dual-Inception [268]	0.732	0.727	0.586	0.566	0.664	0.672	0.862	0.856
STSTNet [123]	0.735	0.760	0.658	0.681	0.680	0.701	0.838	0.868
ME-Recognizer [130]	0.788	0.782	0.775	0.715	0.746	0.753	0.829	0.820
ICE-GAN	**0.874**	**0.883**	**0.879**	**0.883**	**0.782**	**0.801**	**0.895**	**0.904**

TABLE 5.2. Comparisons with the latest graph-based and attention-based approaches in terms of F1-score with LOSO single-database evaluation.

Method	CASME II	SAMM
MicroAttention [214]	0.539	0.402
MER-GCN[135]	0.303	0.283
AU-GACN[236]	0.355	0.433
ICE-GAN	**0.585**	**0.623**

facial movements between onset and apex images, the intra-class information is ignored for each subject.

Our model also achieves the highest scores for individual dataset evaluation. For SAMM, ICE-GAN overpasses ME-Recognizer by 13.4% in UF1 and 23.5% in UAR. The performance in CASME II has improved by 8.0% and 10.2% for UF1 and UAR respectively. However, the improvement on SMIC is not as large as the previous two datasets because there is no clear annotation about onset or apex frames, which cannot give representative information about the subject.

We compare our method with the latest SoTA works following SDE. The model performance of F1-score for 3-category classification is reported in Table 5.2. Compared to MicroAttention [214], since we capture the long-ranged interactions between different facial regions based on graph reasoning, our method achieves a higher F1-score with 8.5% improvement on CASME II and 54.0% on SAMM. MER-GCN [135] and AU-GACN [236] are graph-based methods that reason over AU nodes, while our ICE-GAN achieves a better performance with F1-scores improved by 64.8% and 43.9% compared to AU-GACN, which indicates that our way of constructing graphs over intermediate feature channels can give more representative relational information between facial attributes than directly reason over AU graphs.

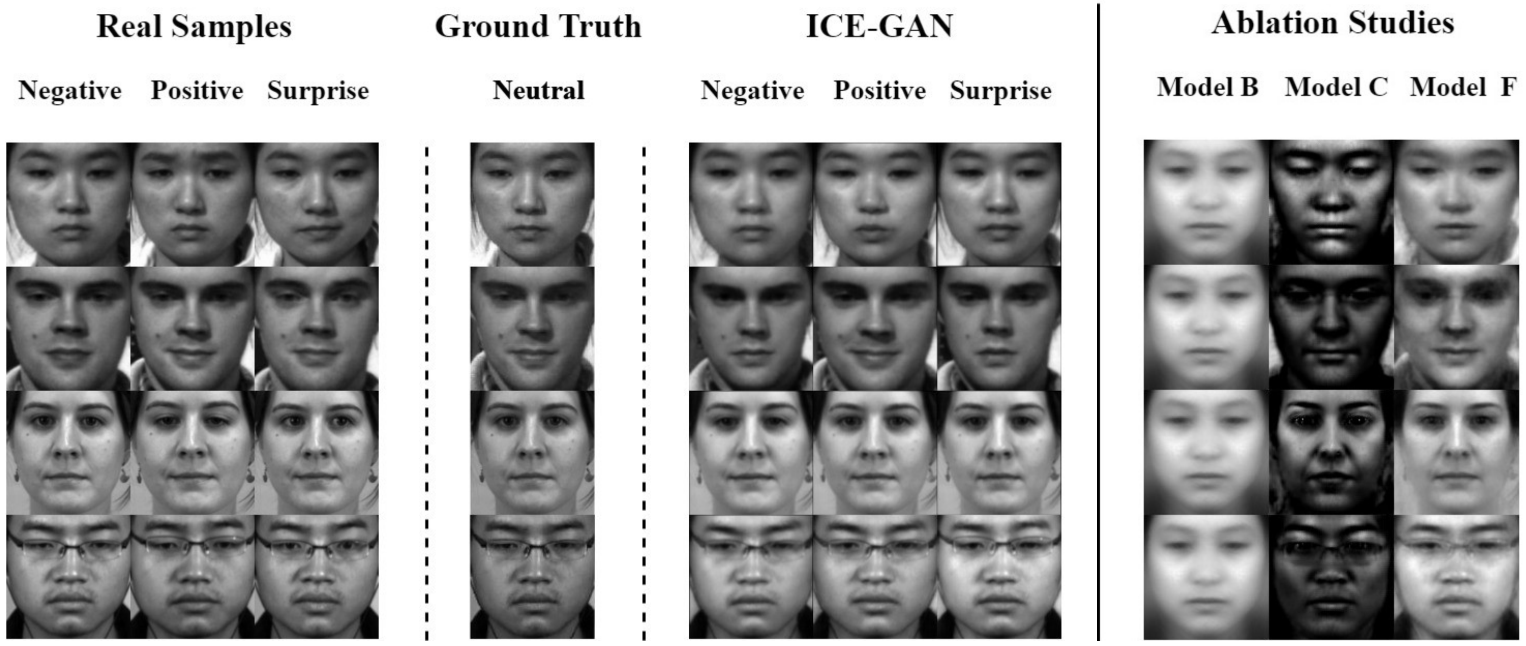
5.4.5 Qualitative Analysis of MES

Generated Video frames are shown in Figure 5.9a, where comparisons between synthetic and real images of four different subjects are demonstrated. We only show the peak frame here instead of the whole video frames as it can be hard to distinguish the difference between adjacent frames when given the whole video. The results indicate that the generated faces achieve no significant artifacts and are basically at the same level as real samples regarding the authenticity, where light conditions are preserved in the images as well.

To better examine synthetic samples, we implement Norm2-based difference between X_{syn} and X_{on} and focused on AU-related face regions. Figure 5.10 presents activated regions shown in red boxes for positive and surprising faces, while negative expressions have more than one kind of MEs and therefore their patterns are difficult to be visualized explicitly. Practically, facial expressions relate to many muscle movements, so multiple AUs can be activated simultaneously. For positive classes, green boxes drawn around the eye area indicate the activation of AU6, and the ones around the lip corner indicate AU12 and AU25. For surprising samples, AU1, AU2, and AU5 are activated, which are all prototypical AUs critical for the recognition of positive (i.e., happy) and surprise micro-expressions, according to FACS [44].

5.4.6 Ablation Study

Extensive experiments are implemented to validate the component design of ICE-GAN. The following studies were examined on the full dataset including SMIC, CASME II, and SAMM, which was further randomly split into a training set with 48 subjects and a testing set with 20 subjects.



(A)

(B)

FIGURE 5.9. (a) Comparisons between real samples and generated samples by ICE-GAN for four different subjects. Neutral images are listed for reference. (b) Synthetic images X_{syn} generated by different model designs.

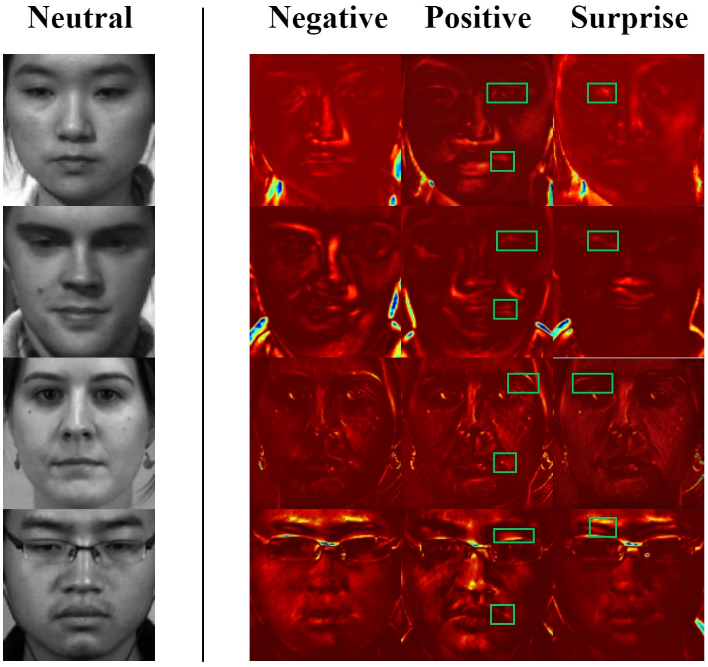


FIGURE 5.10. Norm2-based difference between X_{syn} and X_{on}. Green boxes indicate the subtle muscle movements associated with action units.

TABLE 5.3. Experimental analyses of G and GRM designs on the full dataset split in a subject-wise manner. SC, SE, and GR represent the skip connection, squeeze-and-excitation, and graph reasoning.

Model	D	G_{dec}	G_{enc}	SC	SE	GR	UAR	UF1
A	✓						0.425	0.423
B	✓	✓					0.651	0.660
C	✓	✓	✓				0.705	0.707
D	✓	✓	✓	✓			0.717	0.724
E	✓	✓	✓		✓		0.719	0.732
F	✓	✓	✓			✓	0.761	0.769

Analysis of Generator. Three models are examined to verify the effectiveness of our generator in Table 5.3. Model *A* is the baseline that just uses a discriminative model for MER. Model *B* includes a DCGAN-like generator [169], where UAR and UF1 are increased by 53.2% and 56.0% compared to model *A*, which validates the usefulness of GAN to expand the data size. Model *C* achieves UAR of 0.705 and UF1 of 0.707 based on an encoder-decoder like structure, which proves the advantage of encoding expressive representations of the input data.

Analysis of Graph Reasoning Module. To validate the excellence of the graph-based reasoning module, we compare its performance with skip connections (model *D*) and squeeze-and-excitation (SE) module [215] (model *E*) in Table 5.3. Skip connections simply propagate low-level multi-scale spatial information from encoder to decoder side, however, there is no relation learning or reasoning on latent feature space. Thus, model *D* can only achieve UAR of 0.717 and UF1 of 0.724. Results shown in model *E* indicate that SE module helps the model modulate the channel interdependencies, and model capability is increased compared to model *D* with only skip connections. Moreover, by constructing graphs based on feature channels and learning global relations on local facial features, the model performance has been further improved in the final design (model *F*). Our model achieves the best performance with UAR of 0.761 and UF1 of 0.769, which indicates that relational information between different facial features reasoned from local responses contributes significantly when generating unseen faces.

Analysis of Synthesis Quality. Moreover, synthetic image qualities from models *B*, *C*, and *F* are visualized in Figure 5.9b. The differences between models *B* and *C* demonstrate that finer identity-related attributes can be preserved via the encoder-decoder architecture. With the help of graph-based global reasoning, more high-frequency signals can be passed smoothly through multi-scale connections (e.g., light condition) and artifacts are largely reduced.

Analysis of Discriminator Design. We then exploit the impact of the dimension d_{exp} of *ExpCaps* on MER, within a range of [8, 16, 32, 64, 128]. As observed in Table 5.4, we obtain the best performance when setting d_{exp} to 32, with UAR of 0.761 and UF1 of 0.769. Besides, we compare the capsule-enhanced design with its CNN-based counterparts by replacing the two-layer capsule network with a two-layer CNN, while maintaining the architecture of *PatchGAN*. The performance of CNN-based discriminator is reported in the first row in Table 5.4. To make a fair comparison of the difference in terms of model size, we increased the number of neurons in the two-layer CNN and examined the performance of the enlarged CNN-based discriminator (second row) on the MER task. As reported in Table 5.4, our capsule-enhanced discriminator outperforms its CNN counterparts by a large margin in terms of UAR and UF1, validating that translation equivariance introduced by capsule helps MER.

TABLE 5.4. Ablation studies on the discriminator design. D_{CNN} and D_{CE} denote CNN-based and capsule-enhanced discriminators.

Method	UAR	UF1	#Params
D_{CNN}	0.356	0.321	6.7 MB
D_{CNN} with comparable size	0.432	0.455	85.6 MB
D_{CE} with $d_{exp} = 8$	0.730	0.740	85.7 MB
D_{CE} with $d_{exp} = 16$	0.704	0.723	85.9 MB
D_{CE} with $d_{exp} = 32$	**0.761**	**0.769**	86.4 MB
D_{CE} with $d_{exp} = 64$	0.726	0.732	87.5 MB
D_{CE} with $d_{exp} = 128$	0.713	0.730	89.6 MB

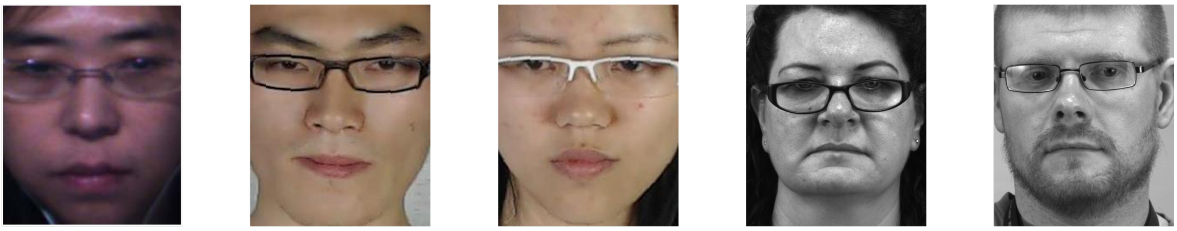


FIGURE 5.11. Failure cases when dealing with SMIC, CASME II, and SAMM datasets.

5.5 Discussion

In this work, we propose to generate face videos with micro expressions by using a conditional GAN network. Although our model performance on face expression recognition has achieved good results on all datasets mentioned in this work (i.e., SMIC, CASME II, and SAMM), our proposed model still fails when encountering face videos where the involved human is wearing eyeglasses. Some failed examples are shown in Figure 5.11, we can see that our model can easily fail and produce bad results when the region of eyes are largely covered by the eyeglasses, hence future work is required to ignore the overlapping objects and put more focus on the face regions.

5.5.1 Ethical Concerns

The application of deep learning in micro-expression recognition, especially in sensitive areas like lie detection and disease diagnosis, raises several ethical concerns, primarily related to privacy and consent. Potential ethical concerns and the corresponding solutions are listed below:

(1) Privacy Protection: Ensuring the privacy of individuals whose facial video data is used is paramount. This involves anonymizing data, where personal identifiers are removed. Techniques like differential privacy, where the data is altered slightly to prevent the identification of individuals, are also used.

(2) Informed Consent: It is crucial to obtain informed consent from participants whose data is used in research. This means they are fully aware of how the data will be used, the purpose of the research, and the potential implications. In some cases, especially in public datasets or surveillance applications, obtaining individual consent might be challenging, and researchers need to navigate these complexities ethically.

(3) Bias and Fairness: Deep learning models can be biased based on the data they are trained on. Researchers need to make sure that the datasets are diverse and representative to avoid biased outcomes, which can have serious implications, especially in lie detection.

5.6 Summary

In this chapter, we propose a model framework which consists of an identity-aware generator with global graph reasoning on local channel graphs for micro-expression synthesis and a capsule-enhanced discriminator for micro-expression recognition. We design a generator that encodes distinguishable facial attributes with side information to control expressions and synthesizes realistic samples by a graph reasoning module, where a channel graph is established via self-attention and global relations between long-ranged facial features are captured. Furthermore, we present a discriminator which improves recognition ability by capturing part-based position-specific face characteristics. Experiments on several datasets demonstrate that our method outperforms SoTA methods by a large margin.

CHAPTER 6

Deep Learning-based Analysis for Text-Guided Face Video Generation

Label-conditioned face video generation has been examined for quite a long time, with variants of GAN-based models generating realistic face videos. However, rather than generating face videos using labels, we discuss that the model ability to create realistic videos from textual descriptions is also important, since texts act as a natural way to explicitly control the generated video content. However, text-to-video generation still faces many challenges: current video datasets have no corresponding textual descriptions, which hinders the development of the designing deep learning-based methods.

In this chapter, we explore the importance of the face video generation in the context of 3D data analysis, and how novel deep learning-based methodologies can be employed to harness this type of data more effectively. We propose a new large-scale facial text-video dataset to assist designing deep learning-based methods for face video generation guided by textual descriptions. Moreover, we propose a novel method based on BERT [53] for face video generation in a zero-shot manner, where the generated face videos are semantically aligned to the input texts as well as consistent in the temporal domain. Extensive experiments validate that our text-guided generation model can effectively produce realistic and consistent face videos, with detailed controls offered by input texts.

6.1 Introduction

Text-guided video generation has recently gained significant attention in the fields of computer vision and computer graphics. By using text as input, video content can be generated and controlled, inspiring numerous applications in both academia and industry [119, 13, 171, 151]. However, text-to-video generation still faces many challenges, particularly in the face-centric scenario where generated video frames often lack quality [68, 131, 119] or have weak relevance to input texts [12, 272, 141, 3]. We believe that one of the main issues is the absence of a well-suited facial text-video dataset containing high-quality video samples and text descriptions of various attributes highly relevant to videos.

This chapter presents **CelebV-Text**, a large-scale, diverse, and high-quality dataset of facial text-video pairs, to facilitate research on facial text-to-video generation tasks. CelebV-Text comprises 70,000 in-the-wild face video clips with diverse visual content, each paired with 20 texts generated using the proposed semi-automatic text generation strategy. The provided texts are of high quality, describing both static and dynamic attributes precisely. The superiority of CelebV-Text over other datasets is demonstrated via comprehensive statistical analysis of the videos, texts, and text-video relevance. The effectiveness and potential of CelebV-Text are further shown through extensive self-evaluation. A benchmark is constructed with representative methods to standardize the evaluation of the facial text-to-video generation task.

The main contributions of this work are summarized as follows:

(1) We propose CelebV-Text, the first large-scale facial text-video dataset with high-quality videos, as well as rich and highly-relevant texts, to facilitate research in facial text-to-video generation.

(2) Comprehensive statistical analyses are conducted to examine video/text quality and diversity, as well as text-video relevance, demonstrating the superiority of CelebV-Text.

(3) A series of self-evaluations are performed to demonstrate the effectiveness and potential of CelebV-Text.

(4) A new benchmark for text-to-video generation is constructed to promote the standardization of the facial text-to-video generation task.

6.2 Literature Review

6.2.1 Diffusion Models

A diffusion model is a type of generative model that captures the joint distribution of data in a stochastic process. The goal is to learn the data distribution, which can then be used to sample new data points. This is done by transforming a simple noise distribution into the data distribution using a series of learned transformations.

The overall diffusion process is illustrated in Figure 6.1. Formally, we can consider a dataset $X = \{x_1, x_2, \ldots, x_n\}$ drawn from an unknown distribution $P(x)$. The diffusion model starts from a known simple distribution P_0, typically a standard Gaussian distribution, and transforms this distribution to match $P(x)$ through a series of noise adding (forward diffusion) and denoising steps (reverse diffusion) [191]. The forward diffusion process is formulated as follows:

$$x_t = \sqrt{1-\beta_t} \cdot x_{t-1} + \sqrt{\beta_t} \cdot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I), \tag{6.1}$$

where x_t is the noisy data at time step t, x_{t-1} is the data from the previous step, ε is noise sampled from a standard multivariate Gaussian, and β_t is a noise schedule hyperparameter that determines how much noise to add at each step. The reverse diffusion process is formulated as:

$$q(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu(x_t, t), \sigma(x_t, t)^2 \cdot I), \tag{6.2}$$

where $\mu(x_t, t)$ and $\sigma(x_t, t)$ are the mean and standard deviation of the denoised data, predicted by U-Net [177]. During training, the model optimizes the parameters of the U-Net to maximize

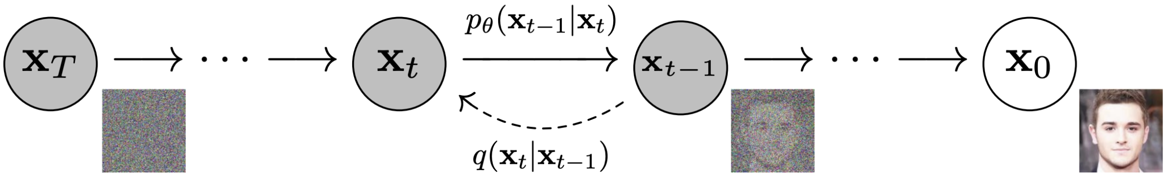

$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_{t-1}$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_0$

FIGURE 6.1. Diffusion model process cited from [80].

the log-likelihood of the data:

$$L = \sum_t \log q(x_{t-1}|x_t). \tag{6.3}$$

In the inference stage, the model generates new data by sampling from the simple distribution P_0 and transforming the sample according to the learned transition operator [80].

6.2.2 Text-to-Video Generation

Text-driven video generation, which involves generating videos from text descriptions, has recently gained significant interest as a challenging task. Mittal *et al.*. [151] first introduced this task to generate semantically consistent videos conditioned on encoded captions. Other studies, such as [52, 13, 161], attempt to generate video samples conditioned on encoded text inputs. However, due to the low richness of text descriptions and the small number of data samples, the generated video samples are often at low resolution or lack relevance with the input texts.

More recently, several works [226, 69, 212, 84, 85, 227, 120] have employed discrete latent codes [58, 159] for more realistic video generation. Some of these works treat videos as a sequence of independent images [227, 120, 84, 69], while Phenaki [212] considers temporal relations between each frame for a more robust video decoding process. Another branch of studies leverage diffusion models for text-to-video generation [82, 70, 79, 190], which require millions or billions of samples to achieve high-quality generation.

While text-to-video generation methods are rapidly evolving, they are generally designed for generating general videos. Among these methods, only MMVID [69] has conducted specific experiments with face-centric descriptions. One possible reason for this is that facial text-to-video generation requires more accurate and detailed text descriptions than general tasks. However, there is currently no suitable dataset available that provides such properties for face-centric text-to-video generation.

6.2.3 Multimodal Datasets

Existing multimodal datasets can be categorized into two classes: open-world and closed-world. Open-world datasets [122, 185, 188, 32, 8, 240, 108, 269, 146, 47, 125, 151] are widely used for text-to-image/video generation tasks. Some of them have manual annotations [122, 188, 240, 108, 47] and part of them are directly collected from the Internet, such as subtitles [185, 146]. Closed-world datasets are mostly composed of images or videos collected in constrained environment with corresponding information such as text. CLEVR [95] is a synthetic text-image dataset produced by arranging 3D objects with different shapes under a controlled background. While MUGEN [72] is a video-audio-text dataset that was collected using CoinRun [42] by introducing audio and new interactions. The corresponding text is produced by human annotators and grammar templates.

Multimodal face datasets also exist. Modified MUG [2] is a closed-world text-video dataset that contains 1,039 videos with subjects showing different emotions, where the text descriptions are generated from facial emotions using a fixed template [102]. MM-Vox [69] contains 19,522 face videos from VoxCeleb [155], with 36 facial attributes manually labeled following CelebA [134] and text descriptions generated via Probabilistic Context-Free Grammar (PCFG) [234]. However, both datasets only contain language descriptions related to static facial attributes without considering the temporal state change (i.e., emotion or action) presented in the original face videos. Moreover, the limited label annotations restrict the diversity of the text descriptions, making them sub-optimal for studying the text-to-video generation task on the face domain. CelebV-HQ [270] is the latest high-quality face video dataset that covers facial annotations, including appearance, movement, and emotion. However, it only provides discrete labels and timestamps, with no text descriptions.

6.3 Methods

In this part, we propose an efficient pipeline, as shown in Figure 6.2, to construct CelebV-Text, including Data Collection & Processing, Data Annotation, and Semi-auto Text Generation.

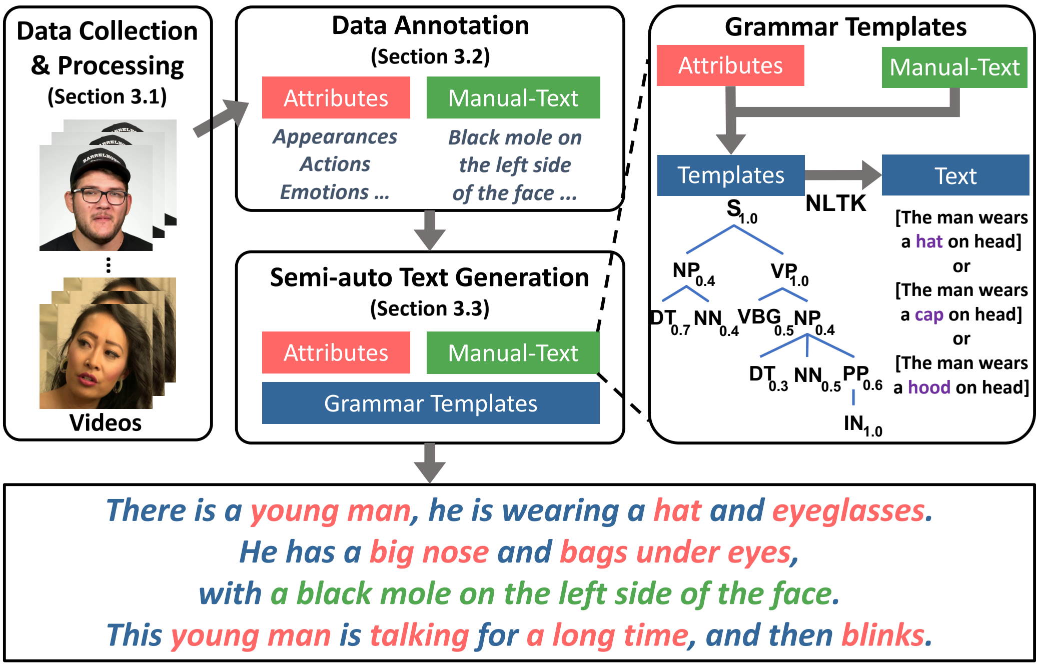

Data Collection & Processing (Section 3.1)
Videos
Data Annotation (Section 3.2)
Attributes
Manual-Text
Appearances Actions Emotions ...
Black mole on the left side of the face ...
Semi-auto Text Generation (Section 3.3)
Attributes
Manual-Text
Grammar Templates
Grammar Templates
Attributes
Manual-Text
Templates
Text
NLTK
S1.0
NP0.4
VP1.0
DT0.7
NN0.4
VBG0.5
NP0.4
DT0.3
NN0.5
PP0.6
IN1.0
[The man wears a hat on head]
or
[The man wears a cap on head]
or
[The man wears a hood on head]
There is a young man, he is wearing a hat and eyeglasses.
He has a big nose and bags under eyes,
with a black mole on the left side of the face.
This young man is talking for a long time, and then blinks.

FIGURE 6.2. **Pipeline of our dataset construction process.** The pipeline includes data collection & processing, data annotation, and semi-auto text generation.

6.3.1 Data Collection & Processing

Collection. We follow the same strategy as CelebV-HQ [270] due to its effectiveness in large-scale high-quality data collection. Specifically, we firstly generate a large number of queries, including human names, movie titles, vlogs and so on, to retrieve videos that contain human faces with temporally dynamic state changes and abundant facial attributes. Our data are collected from open world with videos downloaded from online resources. Videos with low resolution ($< 512^2$), low time duration (< 5s), and having appeared in CelebV-HQ are filtered out.

Processing. To sample high-quality and diverse video clips from our raw collections, similar steps are followed as CelebV-HQ [270] with modifications. We first filter out video clips with bounding box regions less than 512^2 rather than resize them to the same resolution. In this way, clips are not upsampled or downsampled hence the video quality would not be affected, which leads to various resolutions of collected videos: 56.4% with $512^2 \sim 1024^2$, and 43.6% for 1024^2+. To reduce the face area noise when the background changes, we further change the video splitting strategy. In addition to our focus on the same human motion [17] and identity [51] present in adjacent frames, we split the video into different clips when the background changes by a toolkit [1].

6.3.2 Data Annotation

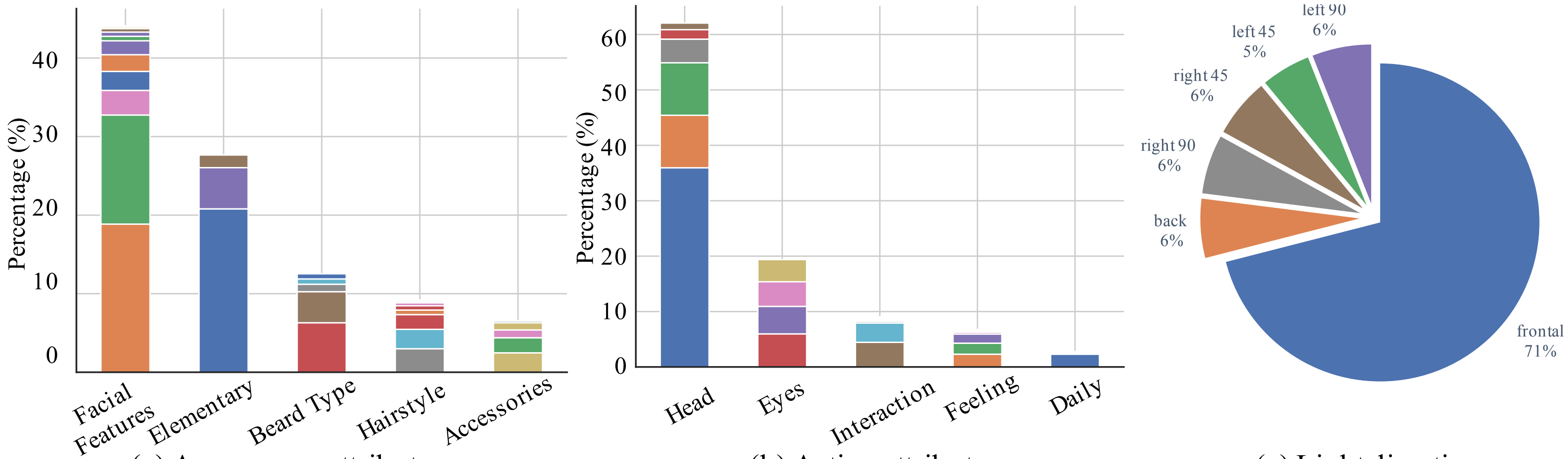
The annotation process is a core part in CelebV-Text construction, which would greatly affect the relevance of text-video pairs, as our designed text templates heavily depend on the annotation results. Here, we first describe how we design attributes, and then give details about the annotation strategy for face videos.

Attributes Design. Temporal dynamic is the key difference between images and videos. However, as shown in Table 6.1, most face video datasets focus on static attributes where attribute information does not change over time, such as appearance. Dynamic attributes that change over time, such as emotion and face actions, are often neglected. In the following, we decouple face videos into static and dynamic categories and details are given as follows.

1) Static. The current dataset [69] only considers static information such as the appearance attribute, which includes 40 classes as CelebA [134]. In contrast, we define static information to include three types of attributes: general appearance, detailed appearance, and light conditions. General appearance attributes follow the same definition as CelebA [134]. Detailed appearance attributes including five classes are proposed for realistic face generation, i.e., scar, mole, freckle, dimple, and one-eyed. We define light conditions in a restricted manner to include light color temperature [77] and brightness [18], with a total of 6 classes.

2) Dynamic. Here, we design three dynamic attributes, i.e., action, emotion, and light directions. For action attributes, we follow CelebV-HQ [270] and expand their action list by two classes, i.e., squint and blink. For emotion attributes, we select the 8 emotion setting in Affectnet [153], including neutral, anger, contempt, disgust, fear, happiness, sadness, and surprise. For light direction attributes, we derive and modify classes from [97] and give 6 light direction classes. Moreover, as shown in Table 6.1, CelebV-HQ [270] is the only

[1] https://github.com/Breakthrough/PySceneDetect



(a) Appearance attributes

(b) Action attributes

(c) Light directions

FIGURE 6.3. **Dataset distribution comparison.** The distributions of appearance attributes, action attributes, and light directions.

dataset giving timestamps of dynamic attributes. Following their idea, we densely annotate all dynamic attributes of CelebV-Text with the start and end time.

Automatic and Manual Annotation. Based on our attributes design, we find that some attributes can be annotated automatically (e.g., appearance) while some need manual annotations (e.g., timestamps of dynamic attributes). Considering the dataset quality and cost of expense, our annotation strategy includes both automatic and manual annotations.

For automatic annotation, we first investigate algorithms and select designed attributes that can be automatically annotated. We then test different algorithms on our dataset and keep those giving annotation accuracy of 85% or higher. This process yields all light condition labels, all appearance labels, and all emotion labels suitable for automatic annotation. Automatic annotation results can be further revised by human workers to improve accuracy in a less costly way.

For manual annotation, we hire and train human workers following [270] to annotate attributes that are filtered out by an automatic annotation process. In this case, we manually annotate dynamic attributes, i.e., action and light directions, to give both class labels and exact timestamps. In addition, it is hard to represent detailed appearance attributes by the discrete label, e.g., the characteristics of scars or moles. We therefore ask annotators to give a natural description for each attribute, describing exact positions relative to face parts. These designs greatly enhance the relevance between the final text and the video.

6.3.3 Semi-auto Text Generation

Multimodal text-video datasets collect texts via three common methods: subtitles [12, 146, 272], manual-text generation [219, 108, 8, 32, 240], and auto-text generation [72, 13, 85]. However, it is difficult for the individual method to generate texts with high relevance to videos, natural expression, and high diversity. Specifically, although subtitles are easy to obtain, they can pose weakly relevant text-video pairs and introduce noise, making the dataset quality hard to control. Moreover, manual-text generation method is time and cost consuming, as natural language descriptions are required for each video. In this case, increasing the data scale is quite hard as more workers are needed to describe new videos, which does not meet the efficiency and scalability of annotation. Finally, auto-text generation is flexible and scalable, as abundant texts can be simultaneously generated given annotation results of collected videos. However, the diversity, complexity, and naturalness of generated texts can be impacted by the designed grammar templates.

To this end, we propose a semi-auto template-based text generation strategy that combines both manual-text and auto-text generation methods. Specifically, as mentioned in Section 6.3.2, manual-texts are required to describe detailed appearance attributes. Annotated attribute information is fed into our designed template for auto-text generation.

To make our template as natural as possible, we first ask each annotator to describe 10 different face videos for each attribute. We then analyze the grammar structure (i.e., parse tree banks) along with online corpora following [106, 30], and find the most three common grammar structures for each attribute. Finally, we utilize probabilistic context-free grammar [193, 234] and modify the grammar structures to design our own templates. Texts are generated based on templates with synonym replacement using NLTK [19] to increase our generation diversity.

6.3.4 MMVID-interp

Due to the difficulty in modelling state change [190, 85], we apply test-time interpolation to MMVID [69], named MMVID-interp, to improve the text encoding and better understand the dynamics. We follow [13] to apply test-time interpolation to MMVID [69] to improve text encoding and better understand the dynamics. Specifically, given the text input describing dynamic attribute changes, we manually split the dynamic description into two sentences, i.e., S_1 and S_2. S_1 contains the description about the appearance and the first dynamic attribute, and S_2 contains the description about the appearance and the second dynamic attribute. Let $\mathbf{t}_{S_1}$ and $\mathbf{t}_{S_2}$ denote the feature representation obtained from the text encoder used in MMVID [127]. In this case, the description about appearance is repeated twice, so that the text encoding of it can be emphasized and improved, making the generation process more stable on preserving face identities. During the sampling process, the encoded text condition $\mathbf{t}$ is obtained by a

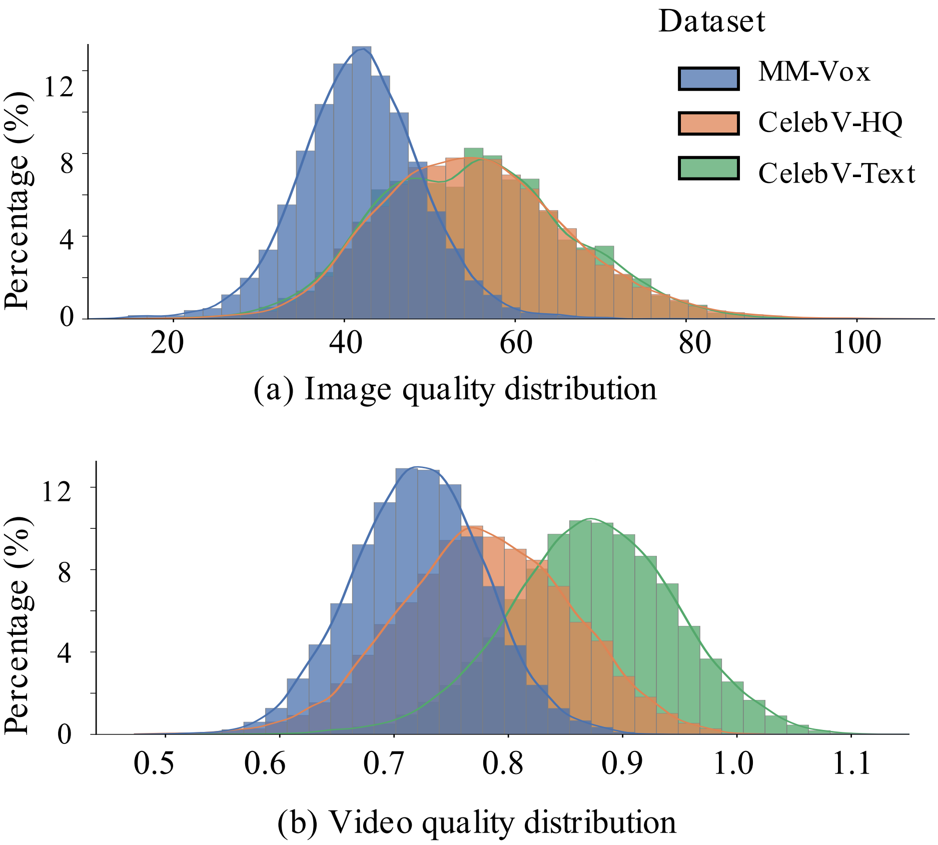


FIGURE 6.4. **Dataset quality distribution.** The metrics used are BRISQUE [150] and VSFA [113] respectively.

linear interpolation between $\mathbf{t}_{S_1}$ and $\mathbf{t}_{S_2}$:

$$\mathbf{t}_i = (1 - \alpha_i)\mathbf{t}_{S_1} + \alpha_i \mathbf{t}_{S_2}, \tag{6.4}$$

where α_i is proportional to the text sequence length. Our modification is simple and will be improved in the future.

6.4 Experiments and Results

6.4.1 Datasets

In this section, we compare our proposed CelebV-Text dataset against existing datasets, i.e., CelebV [230], CelebV-HQ [270] and MM-Vox [69] and so on, in various perspectives.

TABLE 6.1. **In-the-wild face video dataset comparison.** The symbol "#" indicates the number. The abbreviations "Res.", "Dura.", "App.", "Cond.", "Act.", "Emo.", and "Dir." stand for Resolution, Duration, Appearance, Condition, Action, Emotion, and Direction, respectively. The "half checkmark" denotes that CelebV-HQ consists of action attributes with no timestamp.

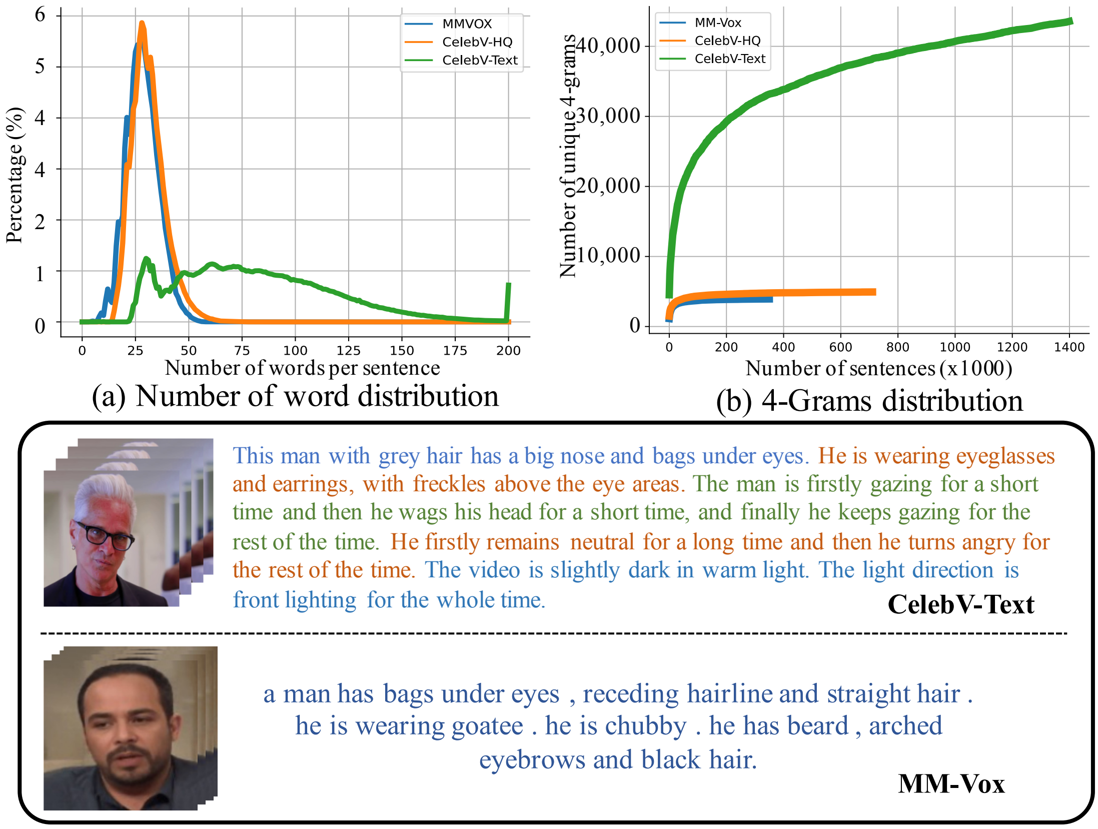
Datasets	Meta Information			Attribute Labels						Text	
				Static			Dynamic				
	#Samples	Res.	Dura.	General App.	Detail App.	Light Cond.	Act.	Emo.	Light Dir.	Auto	Manual
CelebV [230]	5	256×256	2hrs								
VoxCeleb2 [41]	150,480	224×224	2442hrs								
CelebV-HQ [270]	35,666	512× 512	68hrs	✓	✗	✗	✓✗	✓	✗	✗	✗
MM-Vox [69]	19,522	224×224	323hrs	✓	✗	✗	✗	✗	✗	✓	✗
CelebV-Text	70,000	512×512+	279hrs	✓	✓	✓	✓	✓	✓	✓	✓

Video Distribution. We briefly compare the overall statistics of existing face video datasets [230, 41, 270, 69] in Table 6.1. As reported, CelebV-Text contains $70,000$ video clips with a total duration of around 279 hours. Each video is accompanied by 20 sentences describing all 6 designed attributes. Compared to CelebV [230], CelebV-Text has a larger scale and higher resolution. Although VoxCeleb2 [41] has more samples than CelebV-Text, its video distribution is limited as most videos are mainly talking faces. Moreover, video samples of both CelebV-HQ [270] and CelebV-Text are collected in open-world with diverse queries so that they are rich in distribution, while CelebV-Text has about 2 times video data, more video attributes, and highly relevant text descriptions. Finally, compared to the only existing facial text-video dataset MM-Vox [69], CelebV-Text overpasses MM-Vox in terms of scale and quality.

TABLE 6.2. **Multimodal retrieval results.** Clip2Video [60] is leveraged to measure the text-video relevance via retrieval experiments. Bold values indicate the best results, underlined ones indicate the second best.

		Text ⇒ Video					**Video ⇒ Text**				
Description	Dataset	R@1(↑)	R@5(↑)	R@10(↑)	MdR(↓)	MnR(↓)	R@1(↑)	R@5(↑)	R@10(↑)	MdR(↓)	MnR(↓)
(a) App.	MM-Vox [69]	1.5	9.0	15.7	52.0	68.8	2.0	9.2	14.6	43.0	57.8
	CelebV-HQ [270]	5.9	19.2	29.7	27.0	52.2	7.2	20.7	32.4	27.0	46.9
	CelebV-Text	6.1	21.3	35.5	26.3	49.1	7.4	20.7	29.9	26.6	48.3
(b) App.+Emo.	CelebV-HQ [270]	6.5	20.1	30.8	**25.0**	48.0	7.9	25.5	**38.8**	17.0	37.0
	CelebV-Text	6.6	23.4	37.1	26.0	47.6	**8.1**	27.2	34.7	18.2	38.3
(c) App.+Emo.+Act.	CelebV-Text	**6.9**	**24.1**	**39.2**	25.8	**46.7**	8.0	**27.6**	37.1	**16.7**	**36.1**

Attributes Distribution. In order to better present the distribution of different attributes in CelebV-Text, we pick and divide general appearance, action, and light direction attributes



(a) Number of word distribution

(b) 4-Grams distribution

This man with grey hair has a big nose and bags under eyes. He is wearing eyeglasses and earrings, with freckles above the eye areas. The man is firstly gazing for a short time and then he wags his head for a short time, and finally he keeps gazing for the rest of the time. He firstly remains neutral for a long time and then he turns angry for the rest of the time. The video is slightly dark in warm light. The light direction is front lighting for the whole time.

CelebV-Text

a man has bags under eyes , receding hairline and straight hair . he is wearing goatee . he is chubby . he has beard , arched eyebrows and black hair.

MM-Vox

(c) Video and text examples

FIGURE 6.5. **Text distribution.** CelebV-Text achieves better performance in both 4-gram and number words distribution.

into groups. Specifically, all 40 general appearance classes are divided into 5 groups shown in Figure 6.3 (a). Facial features (e.g., double chin, big nose, and oval face) account for the most portion around 45%. The elementary group is twice large than the beard type, accounting for around 25% and 12%, respectively. Fewer samples are located to the hairstyle and accessories groups, taking around 10% and 8%, respectively. Besides, action attributes are divided into 5 groups in Figure 6.3 (b), where it is clear that head-related actions account for the largest portion of around 60%, followed by eyes-related actions of around 20%. The interaction group (e.g., eat), feeling group (e.g., smile), and daily group (e.g., sleep) account for around 9%, 7%, and 4%, respectively. Finally, for light directions (Figure 6.3 (c)), most samples contain the front lighting and the remaining ones are evenly distributed.

TABLE 6.3. **Number of unique POS tags.** The numbers of unique POS tags for MM-Vox, CelebV-HQ, and CelebV-Text.

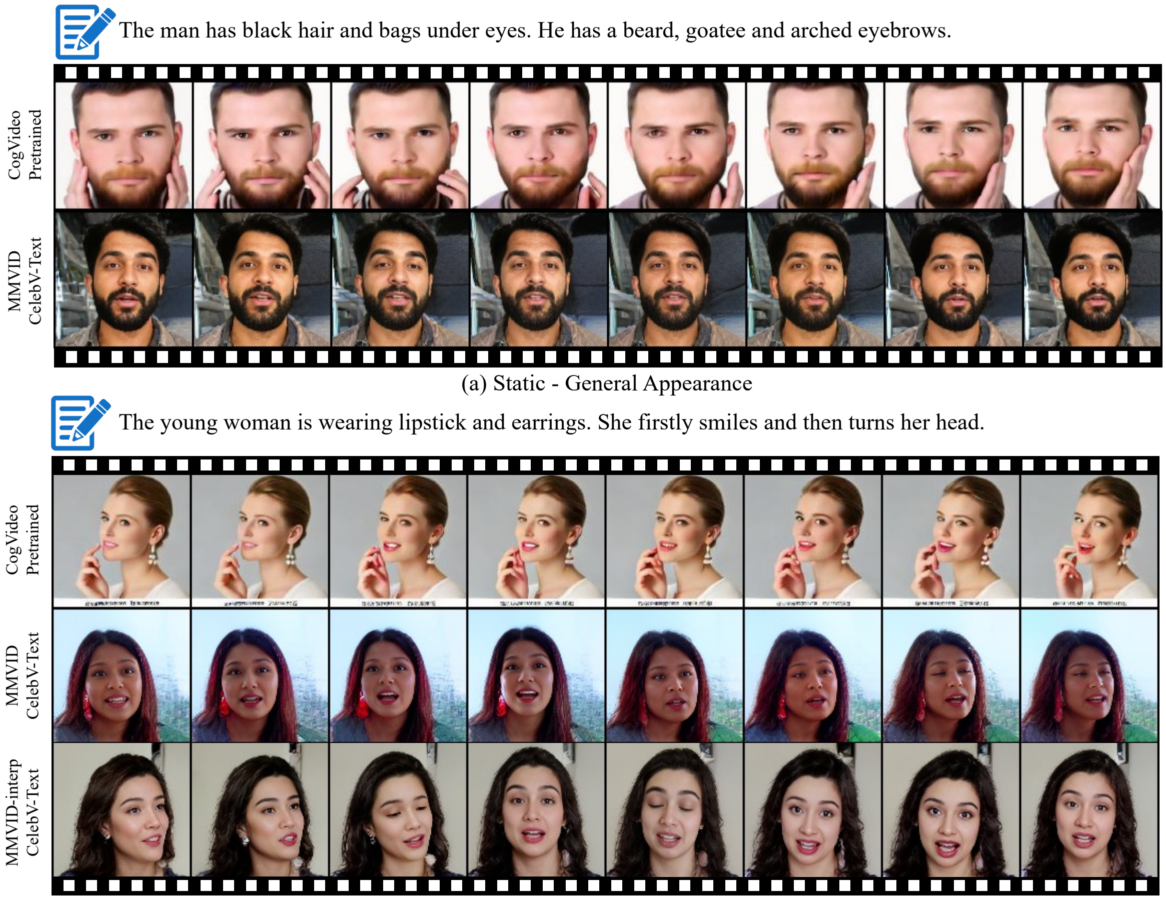
Dataset	#Verb	#Adj.	#Noun	#Adv.
MM-Vox [69]	5	20	38	0
CelebV-HQ [270]	10	24	50	6
CelebV-Text	**96**	**78**	**174**	**24**

Video Quality Distribution. We follow [270] to analyze the quality of our collected videos. To demonstrate the superiority of CelebV-Text, we compare with MM-Vox [69] and CelebV-HQ [270], where mean BRISQUE [150] and VSFA [113] are used to evaluate the image and video quality, respectively. Image quality of all datasets is shown in Figure 6.4 (a), where CelebV-Text and CelebV-HQ achieve comparable quality, higher than MM-Vox by a large margin. Video quality of all datasets is shown in Figure 6.4 (b), where CelebV-Text has the best quality, which is due to the effect of the video split method mentioned in Section 6.3.1, alleviating the discontinuity during background transitions.

Text Comparisons. In addition to a large number of video samples, text descriptions of CelebV-Text are longer and more detailed than those in MM-Vox [69] and CelebV-HQ [270] (see Figure 6.5 (a)), where the average text length of MM-Vox, CelebV-HQ, and CelebV-Text are 28.39, 31.06, and 67.15. Distributions of Celeb-HQ and MM-Vox are close, but there are more words in CelebV-Text to describe a video due to the comprehensive annotation.

To validate the linguistic diversity of the generated texts, comparisons are conducted among the three datasets following [219]. Specifically, we report the unique part-of-speech (POS) tags (i.e., verb, noun, adjective, and adverb) of the three datasets in Table 6.3. Obviously, due to our comprehensively designed attribute list and the number of templates, CelebV-Text presents a wider variety of text styles, covering a broader range of face attributes that are static or dynamic in the temporal domain.

In addition, we further examine the naturalness and complexity of our texts compared to MM-Vox, where we modify [254] to calculate the type-token vocabulary curve for all captions. As shown in Figure 6.5 (b) where unique 4-grams are selected as the types [219], it is evident



The man has black hair and bags under eyes. He has a beard, goatee and arched eyebrows.

(a) Static - General Appearance

The young woman is wearing lipstick and earrings. She firstly smiles and then turns her head.

(b) Dynamic - Action

FIGURE 6.6. **Qualitative results of facial text-to-video generation.** The generated samples are given texts describing (a) the static attribute and (b) dynamic attribute.

that due to our grammar structures and synonym replacement, the linguistic naturalness (vocabulary use) and complexity (vocabulary size) of our CelebV-Text are much better.

6.4.2 Evaluation Metrics

In order to evaluate the performance of different baseline models on our proposed dataset, we apply the FVD [204], FID [78] and CLIPSIM [226] to evaluate the fidelity and generation quality of the generated videos.

For the computation of FVD, suppose we are given a video x, we first use the Inflated 3D ConvNet (I3D) [26] to extract a feature vector $f(x)$. We then fit a multivariate Gaussian distribution to the features of the real videos X and generated videos G, with mean and covariance given by μ_x, Σ_x and μ_g, Σ_g respectively. The FVD between X and G is then defined as the Frechet Distance between these two Gaussian distributions:

$$\mathrm{FVD}(X, G) = ||\mu_x - \mu_g||^2 + \mathrm{Tr}(\Sigma_x + \Sigma_g - 2(\Sigma_x \Sigma_g)^{1/2}), \tag{6.5}$$

where $||.||$ denotes the Euclidean norm, Tr is the trace of a matrix (the sum of the diagonal elements), and $(\Sigma_x \Sigma_g)^{1/2}$ is the matrix square root of the product of the two covariance matrices, which can be computed via the singular value decomposition.

FID is a metric for evaluating the quality of images generated by generative models. In this section, we use FID to examine the per-frame generation quality within the output videos. The mathematical expression for FID is similar to Eq. 6.5.

We follow the same computation of CLIPSIM as [226]:

$$\mathrm{CLIPSIM}(t, \hat{v}) = \frac{1}{N} \sum_{n=1}^{N} \mathrm{CLIP}(t, \hat{v}^{(n)}), \tag{6.6}$$

where t is the input text and $\hat{v}$ is the generated video with N frames. CLIP is the CLIP network from [168].

6.4.3 Implementation Details

We train our network in two separate stages. In the first stage, we train the autoencoder model from VQGAN model [159]. Specifically, we finetune the autoencoder based on our collected image frames, with $f = 16$ (which is the patch size) and $|Z| = 1024$ (which is the vocabulary size of the codebook). In the second stage, we finetune the BERT model using face video data, where we randomly select 16 frames from a given face video and we set the number of the generated frame to be 8. For long-sequence generation output, we apply a frame interpolation tool from Google [173] to interpolate from 8 to 32 frames for presentation.

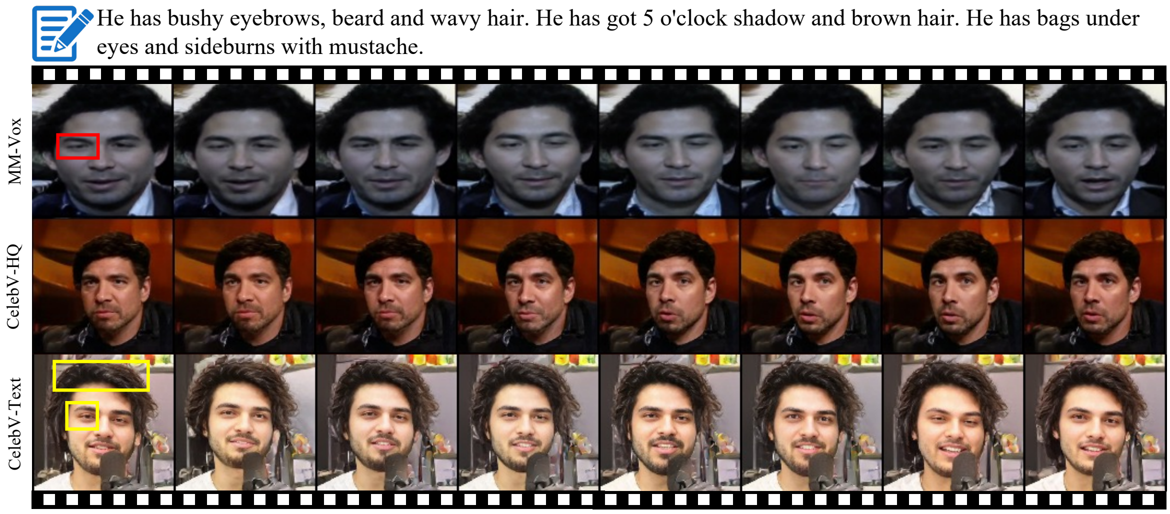

He has bushy eyebrows, beard and wavy hair. He has got 5 o'clock shadow and brown hair. He has bags under eyes and sideburns with mustache.
MM-Vox
CelebV-HQ
CelebV-Text

FIGURE 6.7. **Qualitative results on three facial text-video datasets.** Red and yellow regions indicate the missing of “bags under eyes" and the existence of “wavy hair" and “bags under eyes".

6.4.4 Text-to-Video Generation

To show the benefits brought by our text descriptions which depict both static and dynamic attributes, we conduct experiments to show the effectiveness of CelebV-Text. Experiments are mainly based on a recent open-sourced state-of-the-art method, MMVID [69], and compared with CogVideo[2] [84], which is a large-scale pretrained text-to-video model, trained on millions of text-image/video pairs.

Static Face Video Generation. To validate the effectiveness of our facial text-video dataset in static attributes, we use the models stated above to generate videos conditioned on general appearance, face details, and light conditions descriptions, respectively. Specifically, we first train MMVID [69] from scratch solely on CelebV-Text. We then generate 3 input texts including individual descriptions of each of the static attributes. Generated texts are fed into both MMVID [69] and CogVideo [84] and corresponding video outputs are examined.

Visualization results of general appearance are shown in Figure 6.6 (a), which prove the effectiveness of our dataset. We observe that although CogVideo can output the face video

[2] We choose CogVideo [84] as the representative large-scale model for comparison, since the inference code and pretrained models of other large-scale methods (e.g., CogVideo [84], Phenaki [212], Imagen Video [79], and Make-A-Video[190]) are not public.

given a text description, the text-video pair is not quite relevant, such as “bags under eyes” and “wavy hair”. However, MMVID [69] produces videos with high relevance to input texts, containing all attributes described in the text. More results are shown in Figure 6.8.

Dynamic Face Video Generation. We follow the above experimental setting and leverage MMVID-interp to validate the effectiveness of our dataset with dynamic attribute changes (i.e., emotion, action and light direction). In Figure 6.6 (b), we observe that CogVideo fails to reflect the temporal change described in the input text, i.e., smile → turn. However, both MMVID [69] and MMVID-interp trained on CelebV-Text can successfully model the dynamic attribute changes, which demonstrates the effectiveness of our dataset. In addition, we find that MMVID [69] cannot preserve some attributes well (e.g., earrings), while MMVID-interp can stabilize the sampling process, validating the effectiveness of our modification.

Note that CogVideo [84] has a much larger model size (~ 100 times larger than MMVID [69]) and is trained on much large text-video data (~ 75 times larger than CelebV-Text). However, video samples produced by CogVideo [84] shown in Figure 6.6 are of a lower quality than the ones by MMVID [69] trained solely on CelebV-Text, where generated faces are not in a high relevance to input texts, demonstrating the effectiveness of our facial text-video dataset. More generated video samples with dynamic attribute changes are shown in Figure 6.10 and Figure 6.9.

6.4.5 Benchmarks

As the domain of text-to-video generation is currently thriving, there exists only one benchmark in the face domain, MM-Vox [69]. We expand [69] and construct a benchmark of facial text-to-video generation tasks on three datasets: MM-Vox [69], CelebV-HQ [270] with texts generated by our templates, and CelebV-Text. We choose two representative methods[3], TFGAN [13] and MMVID [69], to evaluate their performances on all datasets.

[3] Other methods, e.g., CogVideo [84], Phenaki [212], Imagen Video [79], and Make-A-Video[190] are not included since their training codes are not public so far.

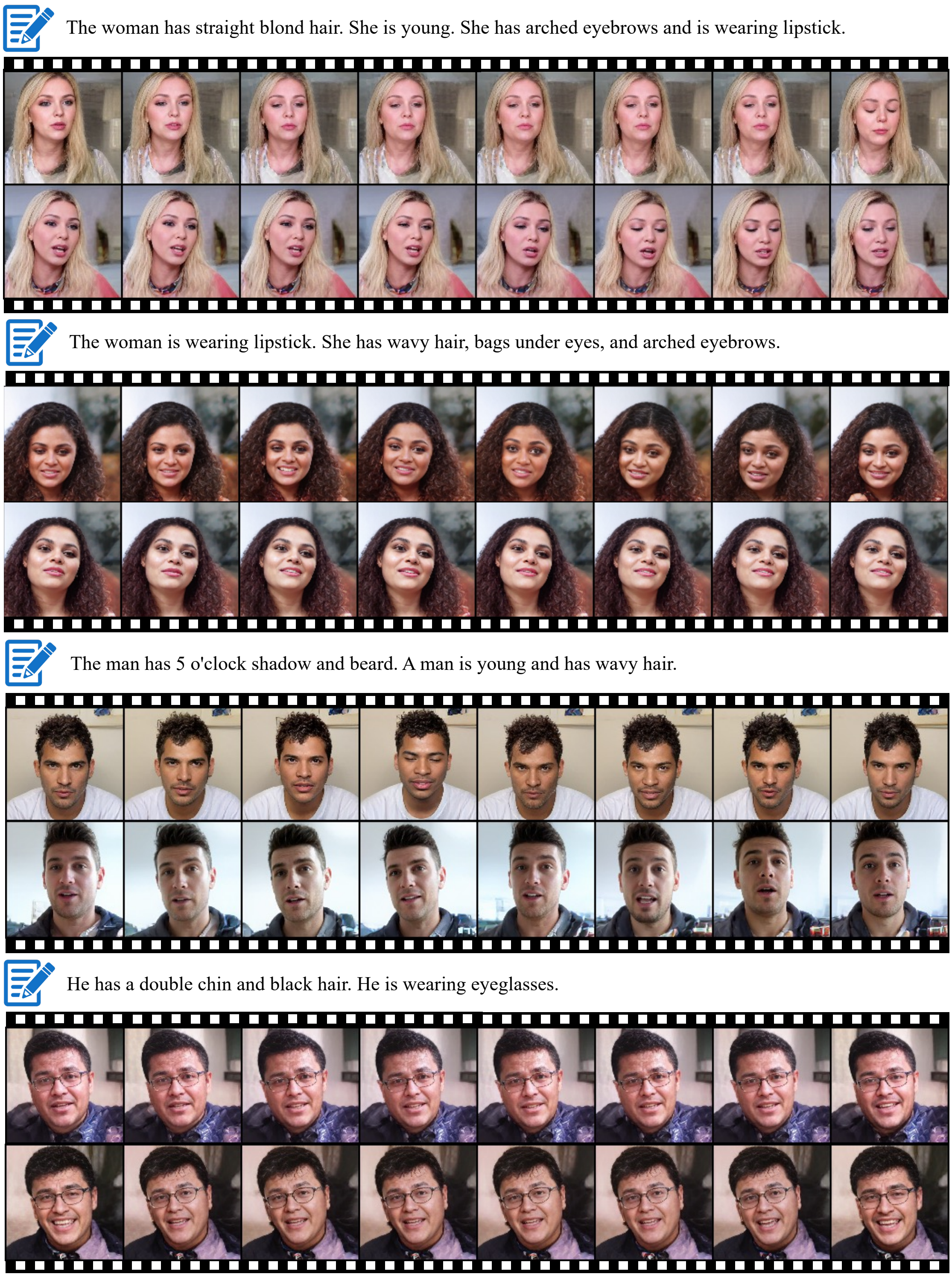
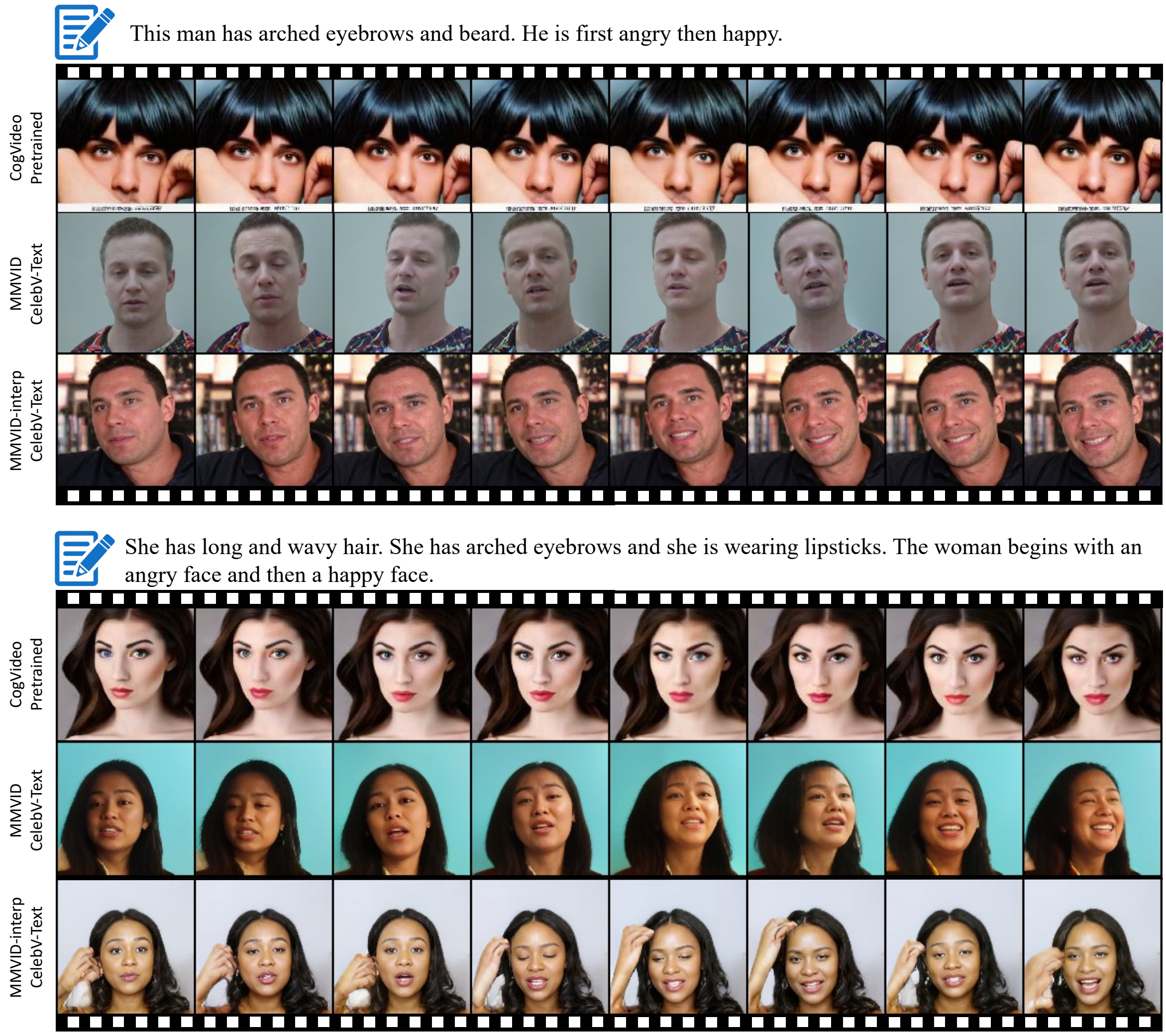

The woman has straight blond hair. She is young. She has arched eyebrows and is wearing lipstick.
The woman is wearing lipstick. She has wavy hair, bags under eyes, and arched eyebrows.
The man has 5 o'clock shadow and beard. A man is young and has wavy hair.
He has a double chin and black hair. He is wearing eyeglasses.

FIGURE 6.8. More sampled results from MMVID with input texts describing general appearances.



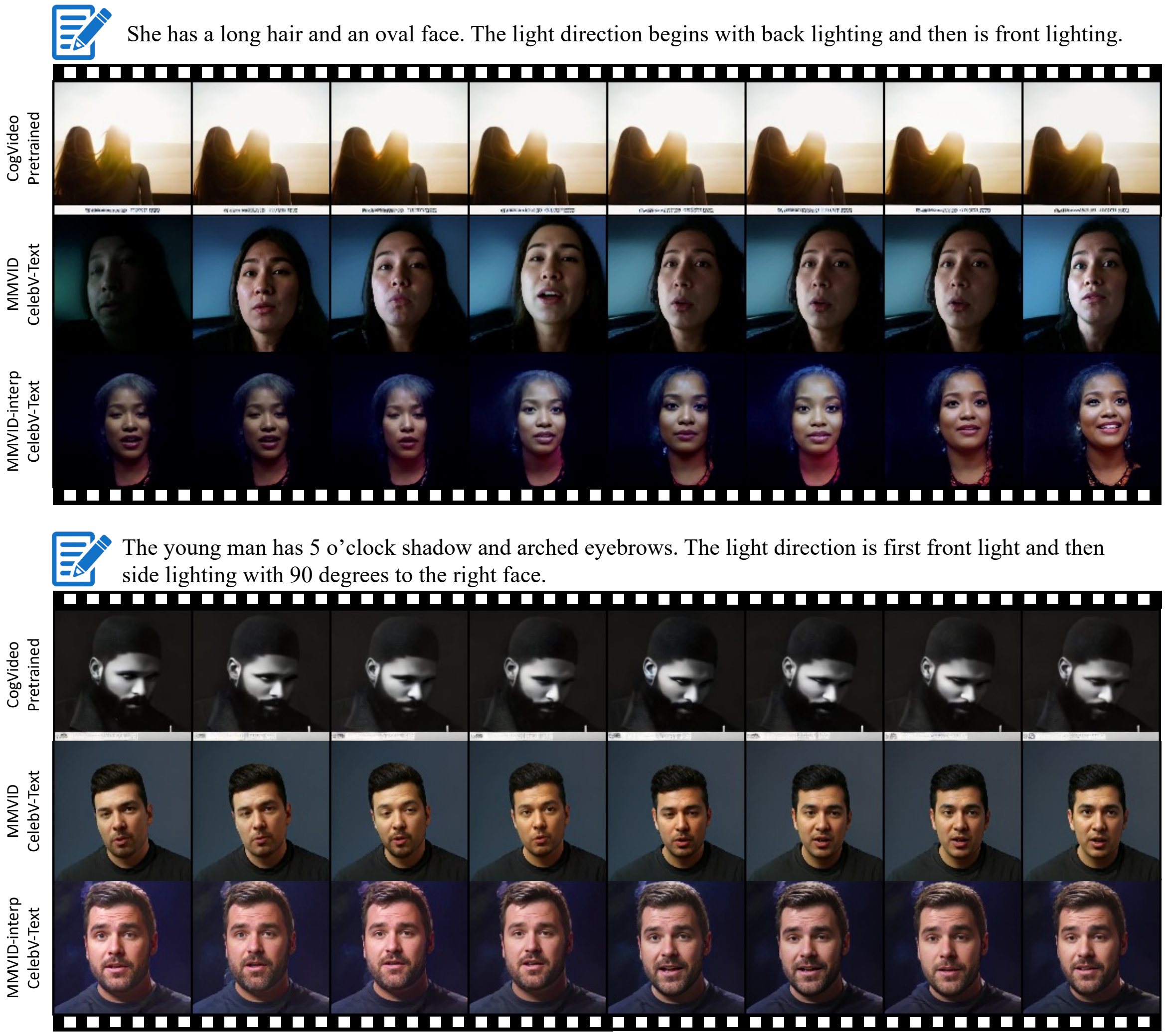
This man has arched eyebrows and beard. He is first angry then happy.

She has long and wavy hair. She has arched eyebrows and she is wearing lipsticks. The woman begins with an angry face and then a happy face.

FIGURE 6.9. **Qualitative results of facial text-to-video generation.** The video samples are generated given texts describing dynamic emotion.

Quantitative Results. For thorough benchmark construction, we evaluate baseline methods given variant texts including static and dynamic attributes. We use FVD [204] (temporal consistency), FID [78] (individual frame quality), and CLIPSIM [226] (text-video relevance) as evaluation metrics following [69] and report detailed results for appearance, action, and emotion in Table 6.4. Evaluation steps are repeated over ten runs with mean values and standard errors reported as well.

It can be seen from Table 6.4 that MMVID [69] obtains good FVD/FID/CLIPSIM metrics over TFGAN [13] which fails to generate reasonable video outputs. In addition, when input


She has a long hair and an oval face. The light direction begins with back lighting and then is front lighting.

The young man has 5 o'clock shadow and arched eyebrows. The light direction is first front light and then side lighting with 90 degrees to the right face.

FIGURE 6.10. **Qualitative results of facial text-to-video generation on dynamic descriptions.** The video samples are generated given texts describing dynamic light directions.

texts contain descriptions about a dynamic state change in the temporal domain, the generated video quality by MMVID [69] decreases, which encourages future methods to focus more on cross-modal understanding and consistent video generation. Moreover, the performance of MMVID-interp is better than MMVID [69] on all metrics, validating the effectiveness of our modification mentioned in Section 6.4.4. Due to challenges posed by our dataset and text-to-video generation task, there is still considerable room to improve.

Qualitative Results. Video samples generated from MMVID [69] trained on different datasets are shown in Figure 6.7, where all video frames are of 128^2. We can see that video samples generated by MMVID [69] trained on different datasets are of high quality with temporal consistency. However, MMVID [69] trained on MM-Vox [69] can sometimes fail to generate attributes mentioned in the input texts.

6.4.6 Ablation Studies

Text-Video Relevance. To quantitatively validate our text-video relevance, we conduct text-video retrieval tasks on three datasets: MM-Vox [69], CelebV-HQ [270], and CelebV-Text. Rather than use conventional frame-wise clip score as most works [212, 79, 190], we follow [60] to compute feature similarities between texts and videos with the consideration of temporal dynamics, which reflects accurate multimodal interactions across the two modalities. Recall at rank K (R@K), median rank (MdR), and mean rank (MnR) [259, 60, 149] are used as evaluation metrics, where the higher R@K, the lower median rank and mean rank indicate better performance.

We first examine the performance given texts with descriptions of general appearance in Table 6.2 (a). Results of CelebV-HQ and CelebV-Text are both better than MM-Vox for two retrieval tasks, which indicates our designed templates can produce texts more relevant to videos than MM-Vox. We further add descriptions about dynamic emotion changes to CelebV-HQ and CelebV-Text in Table 6.2 (b). Similar results are achieved in both datasets, which reflects that our annotation accuracy on static appearance attributes is as good as CelebV-HQ. Finally, we append action descriptions to CelebV-Text in Table 6.2 (c), which achieves the best performance on most metrics, verifying the relevance between our generated texts and video samples.

6.5 Discussion

CelebV-Text can only be used for research purposes. The raw videos will not be released, while the data annotations, links of raw videos, and data processing tools will be released, following a strict legality check procedure of our institution. Note that, our data annotation does not include any personal biometric information (e.g., identity), only generic attribute

TABLE 6.4. **Benchmark of text-to-video generation on different datasets.** ↓ means a lower value is better and ↑ means the opposite.

(A) Quantitative results on general appearance descriptions.

Dataset	Method	FVD(↓)	FID(↓)	CLIPSIM(↑)
MM-Vox [69]	TFGAN [13]	502.28 ± 1.66	760.24 ± 16.01	0.165 ± 0.022
	MMVID [69]	**65.79** ± **1.81**	**38.81** ± **3.66**	**0.170** ± **0.020**
CelebV-HQ [270]	TFGAN [13]	428.04 ± 1.76	616.24 ± 17.45	0.168 ± 0.021
	MMVID [69]	**73.65** ± **1.43**	**63.86** ± **3.66**	**0.172** ± **0.019**
CelebV-Text	TFGAN [13]	403.04 ± 1.34	589.24 ± 16.46	0.177 ± 0.012
	MMVID [69]	**66.69** ± **1.35**	**58.70** ± **4.67**	**0.198** ± **0.014**

(B) Quantitative results on dynamic descriptions of CelebV-Text.

Dataset	Method	FVD(↓)	FID(↓)	CLIPSIM(↑)
CelebV-Text **App.+Emo.**	TFGAN [13]	442.30 ± 2.56	623.17 ± 18.88	0.158 ± 0.024
	MMVID [69]	82.78 ± 1.47	61.58 ± 3.99	0.176 ± 0.008
	MMVID-interp	**72.87** ± **1.23**	**41.57** ± **3.56**	**0.182** ± **0.010**
CelebV-Text **App.+Act.**	TFGAN [13]	571.34 ± 4.54	784.93 ± 20.13	0.154 ± 0.028
	MMVID [69]	109.25 ± 2.11	82.55 ± 4.37	0.174 ± 0.019
	MMVID-interp	**80.81** ± **2.55**	**70.88** ± **4.77**	**0.176** ± **0.020**

information such as gender, hair color, and motion is annotated. Moreover, synthetic videos generated in this chapter do not show bias or certain biometric information (e.g., big lips or big nose), which alleviates the ethical issues. CelebV-Text can be used for deepfakes, while it also can be used for the forgery detection task to prevent this issue. We will strictly control the application and acquisition procedure of CelebV-Text, to avoid possible misuse and abuse. In the future, we will utilize synthetic face generation framework to generate synthetic face videos to overcome the ethical shortcomings of existing real-world face video datasets.

6.5.1 Ethical Concerns

We have to deal with ethical issues when introducing CelebV-Text in the context of potential deepfake creation, and we have to implement controls and ethical safeguards to prevent misuse and ensure responsible use in both research and industry. The plans are listed as follows:

(1) Access Restrictions: Limiting access to the dataset can be a primary control. Researchers might require users to go through an application process, where they

specify their intended use of the data. This process could include background checks or affiliations with recognized institutions.

(2) Use Agreements: Users might be required to sign use agreements or terms of service that explicitly forbid malicious use of the data, such as creating deepfakes for deceptive purposes. Violation of these agreements would have legal consequences.

(3) Ethical Review and Compliance: Ensuring that all uses of CelebV-Text undergo ethical review, especially for sensitive applications like deepfake creation. Compliance with ethical standards and guidelines set by professional organizations or regulatory bodies would be mandatory.

(4) Watermarking and Traceability: Embedding digital watermarks into the dataset can help in tracing the origin of any deepfakes created using CelebV-Text. This aids in accountability and discourages misuse.

(5) Educational and Awareness Initiatives: The research paper might also propose initiatives to educate users about the ethical implications of deepfakes and the importance of responsible usage. This could include workshops, seminars, or online resources.

6.6 Summary

We have proposed CelebV-Text, a large-scale, high-quality, and diverse facial text-video dataset with static and dynamic attributes. CelebV-Text contains $70,000$ video clips, each of which is accompanied by 20 individual sentences describing both static and dynamic factors. Through extensive statistical analysis and experiments, we have demonstrated the superiority and effectiveness of CelebV-Text. In the future, we plan to further enlarge CelebV-Text in both scale and diversity. We may further explore several new tasks based on CelebV-Text, such as fine-grained control of video face, adaptation of general pretrained models to the face domain, and text-driven 3D-aware facial video generation.

CHAPTER 7

Conclusions and Future Work

7.1 Conclusions

In this thesis, we propose to study the urgent problems of different forms of 3D data using deep learning-based methods: point clouds, human meshed, and face videos. The challenges encountered in each form of 3D data and the solutions proposed herein underline the significance and potential of integrating deep learning techniques in 3D data analysis. By examining various disciplines such as computer graphics, virtual reality, and medical imaging, this thesis presents an innovative blend of deep learning techniques to enhance the analysis of 3D data.

We first propose a transformer-base architecture for medical point cloud analysis in Chapter 2. To address the intricate topologies and discrepancies inherent in medical point cloud data, we integrate an enhanced attention module for fast and effective computation and a novel technique, which leverages position embeddings and graph-based reasoning blocks for feature modeling. Our design can effectively tackle the issue posed by limited training samples in medical point clouds, and extensive experiments validate the superiority of our model.

For the analysis of general point cloud data in a real-world scenario with random poses, we propose to extract rotation invariant features from the raw point clouds in Chapter 3. The pipeline of our model design considers rotation invariance as a variant of point cloud registration task and we proposes an effective framework for rotation invariance learning. By defining different geometric regions, such as local and global reference frames, we can extract shape descriptors that are invariant to local and global regions. We further integrate and align the two features by leveraging a transformer architecture. The integrated final feature is ensured to be rotation invariant based on a novel contrastive loss function. By incorporating the our novel learning framework, our method outperforms other advanced methods in point cloud classification, part segmentation and retrieval by a large margin.

The method proposed in Chapter 4 aims at solving the texturing of 3D human mesh models conditioned on textural descriptions. For effective gradient loss direction, we modify SDS and propose a new score function named DSD, which incorporates a negative pair of image and text to iteratively guide the gradient direction during training. In addition, to ensure the generated textures can be semantically aligned to the given geometry, we leverage the geometric depth signal during diffusion to enable the modeling of complex garment details. Furthermore, we propose a learnable network using 3D vertex positions to estimate the BRDF functions and predict the surface material parameters for more accurate texturing. By leveraging physically based rendering techniques and 3D point cloud network, we could generate realistic human avatar textures aligned to input texts, thus successfully producing high-quality 3D human avatars. The conducted user studies indicate that our generated results achieve a better performance than existing SoTA methods.

The final part of our investigation includes face video data. In chapter 5, we introduce a model tailored for video-based micro-expression synthesis and recognition, comprising an identity-aware generator and a capsule-enhanced discriminator. The generator adeptly encodes facial attributes and synthesizes high-quality samples through a graph reasoning module. Meanwhile, our capsule-based discriminator excels in capturing face characteristics vital for micro-expressions. Empirical evaluations on various datasets demonstrate our method's significant superiority over state-of-the-art techniques in the micro-expression domain.

Having explored micro-expression facial data, we transition into face video generation in Chapter 6 and examine the generation task of face videos. The collection and publication of a large-scale face video dataset for the text-to-face generation task mark a significant contribution of this thesis. By proposing a novel application of bidirectional transformers and a new video token training technique, we are able to effectively generate high-quality and consistent face videos conditioned on textual descriptions.

In essence, this thesis serves as a testament to the immense potential of deep learning in 3D data processing. By navigating through point cloud analysis, 3D human modeling, and face video generation, we have proposed solutions that not only augment the computational efficiency and semantic understanding but also enhance the scalability of 3D data analysis techniques. We believe that the strides made in this thesis will open avenues for future research and practical applications in the rapidly evolving domain of 3D data analysis.

7.2 Ethical Concerns

We provide some insights into the challenges and potential strategies related to ensuring the ethical use and limitations of future deep learning methods, especially in the context of generating content based on textual descriptions:

(1) Misinformation and Manipulation: Deep learning models are capable of generating realistic content from textual descriptions can be used to create convincing fake images, videos, or narratives. This poses a significant risk in spreading misinformation or propaganda. To address this, implementing strict usage guidelines, developing detection tools for AI-generated content, and educating users about the potential for such misuse are essential steps. Watermarking AI-generated content can also help distinguish it from authentic human-generated content.

(2) Privacy Concerns: Generating content based on personal data or creating realistic representations of individuals without consent raises privacy issues. To solve this issue, enforcing privacy-preserving practices, such as anonymization or aggregation of data, ensuring data is used ethically, and obtaining explicit consent from individuals whose data is used, are key measures. Compliance with privacy regulations like GDPR is also critical.

(3) Intellectual Property and Copyright: Deep learning models can generate content that infringes on existing intellectual property rights, such as replicating copyrighted material. To solve this, developing algorithms to recognize and respect copyright

boundaries, educating users about intellectual property laws, and implementing filters to prevent copyright violations are essential.

(4) Bias and Fairness: AI models can perpetuate or amplify biases present in their training data, leading to unfair or discriminatory outcomes. Diversifying training datasets, implementing fairness checks, and conducting regular audits to identify and mitigate biases are important. Involving diverse groups in the development process can also help address this issue.

(5) Regulatory Compliance: Keeping up with evolving regulations and ensuring compliance can be challenging for rapidly advancing AI technologies. The potential strategy to address this is to stay informed about regulatory changes, engaging with policymakers, and actively participating in the regulatory discussion are essential for compliance and shaping future guidelines.

7.3 Future Work

We suggest some potential directions using deep learning methods for the challenges that we have mentioned of all three different types of 3D data: point clouds, human meshes, and face videos.

For point cloud analysis, there are two separate ways for medical and rotated point clouds: 1) For medical point clouds, we can combine the information learned from 2D rendered medical images and textual descriptions with the raw 3D point clouds for comprehensive information learning. For example, by leverage the prior information from vision-language pre-training methods such as CLIP [168], recent works [260, 87] are proposing to learn effective point cloud features by using the textural descriptions, rendered 2D images, and 3D points to transfer the CLIP knowledge to 3D vision, which outperforms all previous learning methods; 2) For rotated point clouds, instead of directly learning rotation invariant features, we can disentangle the point clouds for rotation invariant shape and rotation equivariant pose learning. As shown in [33], the disentangled learning of rotation invariance and rotation equivariance allows more flexible model design, resulting in better performances on rotated classification and segmentation tasks.

For human mesh modeling, we propose two directions for future research: 1) Human shape generated conditioned on textual descriptions is a potential task. Although existing methods [83, 91] cannot generate human avatars with detail garment details, their use of the human parameter model [137] opens a way for high-quality human shape generation. A recent method [258] has shown great performance on human avatar generation with detailed clothes conditioned on textual descriptions. We believe that with the help of the differentiable 3D representation [147], more high-quality text-guided human avatars can be generated; 2) With the introduction of large language models and variants of fine-tuned diffusion generative model, human texture generation can be made more realistic and can be fine-grained by detailed textual descriptions.

For face video analysis, we have proposed two methods which leverage GAN and transformer with auto-encoders respectively for face video generation. We believe that using a diffusion model rather than the GAN or auto-encoders can lead to an enhanced generation output. With the development of large language models, the models can be enabled with more accurate text information, which could result in the generated videos more aligned to the input text. Moreover, the text-to-image diffusion generation networks could increase the quality of generated videos. Specifically, CogVideo [84] leverages a private language model for texting encoding, which leads to outstanding generated videos that are semantically aligned to the input texts. Make-A-Video [190] utilizes prior information from text-to-image models to achieve high-quality video generation. Hence, we believe that future researches on language models and diffusion models can enhance the generation quality of face videos.

Bibliography

[1] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna and Ranga Rodrigo. ‘Crosspoint: Self-supervised cross-modal contrastive learning for 3D point cloud understanding’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2022, pp. 9902–9912.

[2] Niki Aifanti, Christos Papachristou and Anastasios Delopoulos. ‘The MUG facial expression database’. In: *Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Services*. IEEE. 2010, pp. 1–4.

[3] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev and Simon Lacoste-Julien. ‘Unsupervised learning from narrated instruction videos’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2016, pp. 4575–4583.

[4] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt and Gerard Pons-Moll. ‘Learning to reconstruct people in clothing from a single RGB camera’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2019, pp. 1175–1186.

[5] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt and Gerard Pons-Moll. ‘Detailed human avatars from monocular video’. In: *Proceedings of the International Conference on 3D Vision*. IEEE. 2018, pp. 98–109.

[6] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt and Gerard Pons-Moll. ‘Video based reconstruction of 3d people models’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2018, pp. 8387–8397.

[7] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt and Marcus Magnor. 'Tex2Shape: Detailed Full Human Body Geometry from a Single Image'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2019, pp. 2293–2303.

[8] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell and Bryan Russell. 'Localizing moments in video with natural language'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2017, pp. 5803–5812.

[9] Martin Arjovsky, Soumith Chintala and Léon Bottou. 'Wasserstein generative adversarial networks'. In: *Proceedings of the International Conference on Machine Learning*. PMLR. 2017, pp. 214–223.

[10] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer and Silvio Savarese. '3D semantic parsing of large-scale indoor spaces'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2016, pp. 1534–1543.

[11] Pietro Astolfi, Ruben Verhagen, Laurent Petit, Emanuele Olivetti, Jonathan Masci, Davide Boscaini and Paolo Avesani. 'Tractogram filtering of anatomically non-plausible fibers with geometric deep learning'. In: *Proceedings of the Medical Image Computing and Computer Assisted Intervention*. Springer. 2020, pp. 291–301.

[12] Max Bain, Arsha Nagrani, Gül Varol and Andrew Zisserman. 'Frozen in time: A joint video and image encoder for end-to-end retrieval'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2021, pp. 1728–1738.

[13] Yogesh Balaji, Martin Renqiang Min, Bing Bai, Rama Chellappa and Hans Peter Graf. 'Conditional GAN with Discriminative Filter Generation for Text-to-Video Synthesis'. In: *Proceedings of the International Joint Conference on Artificial Intelligence*. IJCAI. 2019, pp. 1995–2001.

[14] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla and Pratul P Srinivasan. 'Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2021, pp. 5855–5864.

[15] Irwan Bello. ‘LambdaNetworks: Modeling long-range Interactions without Attention’. In: *Proceedings of the International Conference on Learning Representations*. 2021.

[16] Paul J Besl and Neil D McKay. ‘Method for registration of 3-D shapes’. In: *Sensor fusion*. Vol. 1611. Spie. 1992, pp. 586–606.

[17] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos and Ben Upcroft. ‘Simple online and realtime tracking’. In: *Proceedings of the IEEE International Conference on Image Processing*. IEEE. 2016, pp. 3464–3468.

[18] Sergey Bezryadin, Pavel Bourov and Dmitry Ilinih. ‘Brightness calculation in digital image processing’. In: *Proceedings of the International Symposium on Technologies for Digital Photo Fulfillment*. Vol. 1. Society for Imaging Science and Technology. 2007, pp. 10–15.

[19] Steven Bird, Ewan Klein and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. O’Reilly Media, 2009.

[20] Žiga Bizjak, Boštjan Likar, Franjo Pernuš and Žiga Špiclin. ‘Vascular surface segmentation for intracranial aneurysm isolation and quantification’. In: *Proceedings of the Medical Image Computing and Computer Assisted Intervention*. Springer. 2020, pp. 128–137.

[21] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu and Hendrik Lensch. ‘Nerd: Neural reflectance decomposition from image collections’. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2021, pp. 12684–12694.

[22] Rasmus Bro, Evrim Acar and Tamara G Kolda. ‘Resolving the sign ambiguity in the singular value decomposition’. In: *Journal of Chemometrics* 22.2 (2008), pp. 135–140.

[23] Brent Burley and Walt Disney Animation Studios. ‘Physically-based shading at disney’. In: *Siggraph*. 2012.

[24] Yukang Cao, Yan-Pei Cao, Kai Han, Ying Shan and Kwan-Yee K Wong. ‘Dreamavatar: Text-and-shape guided 3d human avatar generation via diffusion models’. In: *arXiv preprint arXiv:2304.00916* (2023).

[25] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov and Sergey Zagoruyko. ‘End-to-end object detection with transformers’.

In: *Proceedings of the European Conference on Computer Vision*. Springer. 2020, pp. 213–229.

[26] Joao Carreira and Andrew Zisserman. 'Quo vadis, action recognition? a new model and the kinetics dataset'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2017, pp. 6299–6308.

[27] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su et al. 'Shapenet: An information-rich 3D model repository'. In: *arXiv:1512.03012*. 2015.

[28] Chao Chen, Guanbin Li, Ruijia Xu, Tianshui Chen, Meng Wang and Liang Lin. 'Clusternet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2019, pp. 4994–5002.

[29] Chun-Fu Richard Chen, Quanfu Fan and Rameswar Panda. 'Crossvit: Cross-attention multi-scale vision transformer for image classification'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2021, pp. 357–366.

[30] Danqi Chen and Christopher D Manning. 'A fast and accurate dependency parser using neural networks'. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. ACL, 2014, pp. 740–750.

[31] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov and Matthias Nießner. 'Text2Tex: Text-driven texture synthesis via diffusion models'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2023.

[32] David Chen and William B Dolan. 'Collecting highly parallel data for paraphrase evaluation'. In: *Proceedings of the Association for Computational Linguistics*. 2011, pp. 190–200.

[33] Ronghan Chen and Yang Cong. 'The Devil is in the Pose: Ambiguity-free 3D Rotation-invariant Learning via Pose-aware Convolution'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2022, pp. 7472–7481.

[34] Rui Chen, Yongwei Chen, Ningxin Jiao and Kui Jia. 'Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2023.

[35] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng and Yannis Kalantidis. 'Graph-based global reasoning networks'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2019, pp. 433–442.

[36] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li and Bingbing Liu. '(AF)2-S3Net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2021, pp. 12547–12556.

[37] Rewon Child, Scott Gray, Alec Radford and Ilya Sutskever. 'Generating long sequences with sparse transformers'. In: *arXiv preprint arXiv:1904.10509* (2019).

[38] Jaesung Choe, Chunghyun Park, Francois Rameau, Jaesik Park and In So Kweon. 'Pointmixer: Mlp-mixer for point cloud understanding'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2022, pp. 620–640.

[39] Hongsuk Choi, Gyeongsik Moon and Kyoung Mu Lee. 'Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose'. In: *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 769–787.

[40] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell and Adrian Weller. 'Rethinking Attention with Performers'. In: *Proceedings of the International Conference on Learning Representations*. 2021.

[41] Joon Son Chung, Arsha Nagrani and Andrew Zisserman. 'Voxceleb2: Deep speaker recognition'. In: *Proceedings of the Interspeech*. 2018.

[42] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim and John Schulman. 'Quantifying generalization in reinforcement learning'. In: *Proceedings of the International Conference on Machine Learning*. PMLR. 2019, pp. 1282–1289.

[43] Taco S. Cohen, Mario Geiger, Jonas Köhler and Max Welling. 'Spherical CNNs'. In: *Proceedings of the International Conference on Learning Representations*. 2018.

[44] Jeffrey F Cohn, Zara Ambadar and Paul Ekman. ‘Observer-based measurement of facial expression with the Facial Action Coding System’. In: *The Handbook of Emotion Elicitation and Assessment* 1.3 (2007), pp. 203–221.

[45] R. L. Cook and K. E. Torrance. ‘A Reflectance Model for Computer Graphics’. In: *ACM Transactions on Graphics* 1.1 (1982), pp. 7–24.

[46] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser and Matthias Nießner. ‘Scannet: Richly-annotated 3d reconstructions of indoor scenes’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 5828–5839.

[47] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price et al. ‘Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100’. In: *International Journal of Computer Vision* (2022), pp. 1–23.

[48] Adrian K Davison, Cliff Lansley, Nicholas Costen, Kevin Tan and Moi Hoon Yap. ‘Samm: A spontaneous micro-facial movement dataset’. In: *IEEE Transactions on Affective Computing* 9.1 (2016), pp. 116–129.

[49] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi and Leonidas J Guibas. ‘Vector neurons: A general framework for SO (3)-equivariant networks’. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2021, pp. 12200–12209.

[50] Haowen Deng, Tolga Birdal and Slobodan Ilic. ‘PPFNet: Global context aware local features for robust 3d point matching’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2018, pp. 195–205.

[51] Jiankang Deng, Jia Guo, Niannan Xue and Stefanos Zafeiriou. ‘Arcface: Additive angular margin loss for deep face recognition’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2019, pp. 4690–4699.

[52] Kangle Deng, Tianyi Fei, Xin Huang and Yuxin Peng. ‘IRC-GAN: Introspective Recurrent Convolutional GAN for Text-to-Video Generation’. In: *Proceedings of the International Joint Conference on Artificial Intelligence*. IJCAI. 2019, pp. 2216–2222.

[53] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. ‘BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding’. In: *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Vol. 1. ACL, 2019, p. 2.

[54] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko and Trevor Darrell. ‘Long-term recurrent convolutional networks for visual recognition and description’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2015, pp. 2625–2634.

[55] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit and Neil Houlsby. ‘An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale’. In: *Proceedings of the International Conference on Learning Representations*. 2021.

[56] Bertram Drost, Markus Ulrich, Nassir Navab and Slobodan Ilic. ‘Model globally, match locally: Efficient and robust 3D object recognition’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2010, pp. 998–1005.

[57] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik and Ryan P Adams. ‘Convolutional networks on graphs for learning molecular fingerprints’. In: *Advances in neural information processing systems* 28 (2015).

[58] Patrick Esser, Robin Rombach and Bjorn Ommer. ‘Taming transformers for high-resolution image synthesis’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2021, pp. 12873–12883.

[59] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia and Kostas Daniilidis. ‘Learning SO(3) equivariant representations with spherical CNNs’. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2018, pp. 52–68.

[60] Han Fang, Pengfei Xiong, Luhui Xu and Yu Chen. ‘CLIP2Video: Mastering Video-Text Retrieval via Image CLIP’. In: *arXiv preprint arXiv:2106.11097* (2021).

[61] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic and Sanja Fidler. 'Get3d: A generative model of high quality 3d textured shapes learned from images'. In: *Advances In Neural Information Processing Systems* 35 (2022), pp. 31841–31854.

[62] Shubham Goel, Angjoo Kanazawa and Jitendra Malik. 'Shape and viewpoint without keypoints'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2020, pp. 88–104.

[63] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville and Yoshua Bengio. 'Generative adversarial nets'. In: *Advances in Neural Information Processing Systems* 27 (2014).

[64] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias and Gigel Macesanu. 'A survey of deep learning techniques for autonomous driving'. In: *Journal of Field Robotics* 37.3 (2020), pp. 362–386.

[65] Ruibin Gu, Qiuxia Wu, Yuqiong Li, Wenxiong Kang, Wing WY Ng and Zhiyong Wang. 'Enhanced local and global learning for rotation-invariant point cloud representation'. In: *IEEE MultiMedia* 29.4 (2022), pp. 24–37.

[66] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin and Shi-Min Hu. 'Pct: Point cloud transformer'. In: *Computational Visual Media* 7 (2021), pp. 187–199.

[67] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu and Mohammed Bennamoun. 'Deep learning for 3d point clouds: A survey'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 43.12 (2020), pp. 4338–4364.

[68] Tanmay Gupta, Dustin Schwenk, Ali Farhadi, Derek Hoiem and Aniruddha Kembhavi. 'Imagine this! scripts to compositions to videos'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2018, pp. 598–613.

[69] Ligong Han, Jian Ren, Hsin-Ying Lee, Francesco Barbieri, Kyle Olszewski, Shervin Minaee, Dimitris Metaxas and Sergey Tulyakov. 'Show me what and tell me how: Video synthesis via multimodal conditioning'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2022, pp. 3615–3625.

[70] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach and Frank Wood. 'Flexible Diffusion Modeling of Long Videos'. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 27953–27965.

[71] Nils Hasler, Carsten Stoll, Martin Sunkel, Bodo Rosenhahn and H-P Seidel. 'A statistical model of human pose and body shape'. In: *Computer graphics forum*. Vol. 28. 2. Wiley Online Library. 2009, pp. 337–346.

[72] Thomas Hayes, Songyang Zhang, Xi Yin, Guan Pang, Sasha Sheng, Harry Yang, Songwei Ge, Qiyuan Hu and Devi Parikh. 'Mugen: A playground for video-audio-text multimodal understanding and generation'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2022, pp. 431–449.

[73] Jiafa He, Chengwei Pan, Can Yang, Ming Zhang, Yang Wang, Xiaowei Zhou and Yizhou Yu. 'Learning Hybrid Representations for Automatic 3D Vessel Centerline Extraction'. In: *Proceedings of the Medical Image Computing and Computer Assisted Intervention*. Springer. 2020, pp. 24–34.

[74] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie and Ross Girshick. 'Momentum contrast for unsupervised visual representation learning'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2020, pp. 9729–9738.

[75] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. 'Deep residual learning for image recognition'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 770–778.

[76] Ying He, Su-Jing Wang, Jingting Li and Moi Hoon Yap. 'Spotting macro-and micro-expression intervals in long video sequences'. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE. 2020, pp. 742–748.

[77] Javier Hernandez-Andres, Raymond L Lee and Javier Romero. 'Calculating correlated color temperatures across the entire gamut of daylight and skylight chromaticities'. In: *Applied optics* 38.27 (1999), pp. 5703–5709.

[78] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler and Sepp Hochreiter. ‘Gans trained by a two time-scale update rule converge to a local nash equilibrium’. In: *Advances in Neural Information Processing Systems* 30 (2017).

[79] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet et al. ‘Imagen Video: High Definition Video Generation with Diffusion Models’. In: *arXiv preprint arXiv:2210.02303* (2022).

[80] Jonathan Ho, Ajay Jain and Pieter Abbeel. ‘Denoising diffusion probabilistic models’. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6840–6851.

[81] Jonathan Ho and Tim Salimans. ‘Classifier-free diffusion guidance’. In: *arXiv preprint arXiv:2207.12598* (2022).

[82] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi and David J Fleet. ‘Video diffusion models’. In: *arXiv preprint arXiv:2204.03458* (2022).

[83] Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang and Ziwei Liu. ‘AvatarCLIP: Zero-Shot Text-Driven Generation and Animation of 3D Avatars’. In: *ACM Transactions on Graphics* 41.4 (2022).

[84] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu and Jie Tang. ‘CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformers’. In: *Proceedings of the International Conference on Learning Representations*. 2023.

[85] Yaosi Hu, Chong Luo and Zhenzhong Chen. ‘Make it move: controllable image-to-video generation with text descriptions’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2022, pp. 18219–18228.

[86] Binh-Son Hua, Minh-Khoi Tran and Sai-Kit Yeung. ‘Pointwise convolutional neural networks’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018, pp. 984–993.

[87] Tianyu Huang, Bowen Dong, Yunhan Yang, Xiaoshui Huang, Rynson WH Lau, Wanli Ouyang and Wangmeng Zuo. ‘Clip2point: Transfer clip to point cloud classification with image-depth pre-training’. In: *arXiv preprint arXiv:2210.01055* (2022).

[88] Catalin Ionescu, Dragos Papava, Vlad Olaru and Cristian Sminchisescu. 'Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.7 (2013), pp. 1325–1339.

[89] Max Jaderberg, Karen Simonyan, Andrew Zisserman et al. 'Spatial transformer networks'. In: *Advances in Neural Information Processing Systems* 28 (2015).

[90] Shuiwang Ji, Wei Xu, Ming Yang and Kai Yu. '3D convolutional neural networks for human action recognition'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 35.1 (2012), pp. 221–231.

[91] Ruixiang Jiang, Can Wang, Jingbo Zhang, Menglei Chai, Mingming He, Dongdong Chen and Jing Liao. 'AvatarCraft: Transforming Text into Neural Human Avatars with Parameterized Shape and Pose Control'. In: *arXiv preprint arXiv:2303.17606* (2023).

[92] Haifeng Jin, Qingquan Song and Xia Hu. 'Auto-keras: An efficient neural architecture search system'. In: *Proceedings of the SIGKDD International Conference on Knowledge Discovery & Data mining*. ACM, 2019, pp. 1946–1956.

[93] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend and Ron Dror. 'Learning from protein structure with geometric vector perceptrons'. In: *Proceedings of the International Conference on Learning Representations*. 2021.

[94] Justin Johnson, Alexandre Alahi and Li Fei-Fei. 'Perceptual losses for real-time style transfer and super-resolution'. In: *Proceedings of the European Conference on Computer Vision Workshops*. Springer. 2016, pp. 694–711.

[95] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick and Ross Girshick. 'Clevr: A diagnostic dataset for compositional language and elementary visual reasoning'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2017, pp. 2901–2910.

[96] Sarah Jordan, Laure Brimbal, D Brian Wallace, Saul M Kassin, Maria Hartwig and Chris NH Street. 'A test of the micro-expressions training tool: Does it improve lie detection?' In: *Journal of Investigative Psychology and Offender Profiling* 16.3 (2019), pp. 222–235.

[97] Peter Kán and Hannes Kafumann. ‘Deeplight: light source estimation for augmented reality using deep learning’. In: *The Visual Computer* 35 (2019), pp. 873–883.

[98] Brian Karis and Epic Games. ‘Real shading in unreal engine 4’. In: *Physically Based Shading Theory Practice* 4.3 (2013), p. 1.

[99] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas and François Fleuret. ‘Transformers are rnns: Fast autoregressive transformers with linear attention’. In: *Proceedings of the International Conference on Machine Learning*. PMLR. 2020.

[100] Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan et al. ‘Identifying medical diagnoses and treatable diseases by image-based deep learning’. In: *cell* 172.5 (2018), pp. 1122–1131.

[101] Huai-Qian Khor, John See, Raphael Chung Wei Phan and Weiyao Lin. ‘Enriched long-term recurrent convolutional network for facial micro-expression recognition’. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE. 2018, pp. 667–674.

[102] Doyeon Kim, Donggyu Joo and Junmo Kim. ‘Tivgan: Text to image to video generation with step-by-step evolutionary generator’. In: *IEEE Access* 8 (2020), pp. 153113–153122.

[103] Seohyun Kim, Jaeyoo Park and Bohyung Han. ‘Rotation-invariant local-to-global representation learning for 3d point cloud’. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 8174–8185.

[104] Thomas N. Kipf and Max Welling. ‘Semi-Supervised Classification with Graph Convolutional Networks’. In: *Proceedings of the International Conference on Learning Representations*. 2017.

[105] Nikita Kitaev, Lukasz Kaiser and Anselm Levskaya. ‘Reformer: The Efficient Transformer’. In: *Proceedings of the International Conference on Learning Representations*. 2020.

[106] Dan Klein and Christopher D Manning. ‘Accurate unlexicalized parsing’. In: *Proceedings of the Association for Computational Linguistics*. ACL, 2003, pp. 423–430.

[107] Nikos Kolotouros, Thiemo Alldieck, Andrei Zanfir, Eduard Gabriel Bazavan, Mihai Fieraru and Cristian Sminchisescu. 'DreamHuman: Animatable 3D Avatars from Text'. In: *arXiv preprint arXiv:2306.09329* (2023).

[108] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei and Juan Carlos Niebles. 'Dense-captioning events in videos'. In: *Proceedings of the IEEE International Conference on Computer Vision*. IEEE. 2017, pp. 706–715.

[109] Rodney LaLonde and Ulas Bagci. 'Capsules for Object Segmentation'. In: *arXiv preprint arXiv:1804.04241* (2018).

[110] Loic Landrieu and Martin Simonovsky. 'Large-scale point cloud semantic segmentation with superpoint graphs'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2018, pp. 4558–4567.

[111] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi and Yee Whye Teh. 'Set transformer: A framework for attention-based permutation-invariant neural networks'. In: *Proceedings of the International Conference on Machine Learning*. PMLR. 2019.

[112] Jiabao Lei, Yabin Zhang, Kui Jia et al. 'Tango: Text-driven photorealistic and robust 3d stylization via lighting decomposition'. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 30923–30936.

[113] Dingquan Li, Tingting Jiang and Ming Jiang. 'Quality assessment of in-the-wild videos'. In: *Proceedings of the ACM International Conference on Multimedia*. ACM. 2019, pp. 2351–2359.

[114] Feiran Li, Kent Fujiwara, Fumio Okura and Yasuyuki Matsushita. 'A Closer Look at Rotation-Invariant Deep Point Cloud Analysis'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2021, pp. 16218–16227.

[115] Jiaxin Li, Ben M Chen and Gim Hee Lee. 'So-net: Self-organizing network for point cloud analysis'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2018, pp. 9397–9406.

[116] Xianzhi Li, Ruihui Li, Guangyong Chen, Chi-Wing Fu, Daniel Cohen-Or and Pheng-Ann Heng. 'A rotation-invariant framework for deep point cloud analysis'. In: *IEEE Transactions on Visualization and Computer Graphics* 28.12 (2021), pp. 4503–4514.

[117] Xiaobai Li, Tomas Pfister, Xiaohua Huang, Guoying Zhao and Matti Pietikäinen. 'A spontaneous micro-expression database: Inducement, collection and baseline'. In: *Proceedings of the IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*. IEEE. 2013, pp. 1–6.

[118] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di and Baoquan Chen. 'Pointcnn: Convolution on x-transformed points'. In: *Advances in Neural Information Processing Systems* 31 (2018).

[119] Yitong Li, Martin Min, Dinghan Shen, David Carlson and Lawrence Carin. 'Video generation from text'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. AAAI Press. 2018.

[120] Jian Liang, Chenfei Wu, Xiaowei Hu, Zhe Gan, Jianfeng Wang, Lijuan Wang, Zicheng Liu, Yuejian Fang and Nan Duan. 'NUWA-infinity: Autoregressive over autoregressive generation for infinite visual synthesis'. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 15420–15432.

[121] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu and Tsung-Yi Lin. 'Magic3d: High-resolution text-to-3d content creation'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2023, pp. 300–309.

[122] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár and C Lawrence Zitnick. 'Microsoft coco: Common objects in context'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2014, pp. 740–755.

[123] Sze-Teng Liong, Yee Siang Gan, John See, Huai-Qian Khor and Yen-Chang Huang. 'Shallow triple stream three-dimensional cnn (ststnet) for micro-expression recognition'. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE. 2019, pp. 1–5.

[124] Sze-Teng Liong, Yee Siang Gan, Danna Zheng, Shu-Meng Li, Hao-Xuan Xu, Han-Zhe Zhang, Ran-Ke Lyu and Kun-Hong Liu. 'Evaluation of the spatio-temporal features and gan for micro-expression recognition system'. In: *Signal Processing Systems* 92 (2020), pp. 705–725.

[125] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma and Shenghua Gao. ‘Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis’. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2019, pp. 5904–5913.

[126] Xinhai Liu, Zhizhong Han, Yu-Shen Liu and Matthias Zwicker. ‘Point2sequence: Learning the shape representation of 3d point clouds with an attention-based sequence to sequence network’. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. AAAI Press, 2019, pp. 8778–8785.

[127] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer and Veselin Stoyanov. ‘Roberta: A robustly optimized bert pretraining approach’. In: *arXiv preprint arXiv:1907.11692* (2019).

[128] Yongcheng Liu, Bin Fan, Gaofeng Meng, Jiwen Lu, Shiming Xiang and Chunhong Pan. ‘Densepoint: Learning densely contextual representation for efficient point cloud processing’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2019, pp. 5239–5248.

[129] Yongcheng Liu, Bin Fan, Shiming Xiang and Chunhong Pan. ‘Relation-shape convolutional neural network for point cloud analysis’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2019, pp. 8895–8904.

[130] Yuchi Liu, Heming Du, Liang Zheng and Tom Gedeon. ‘A neural micro-expression recognizer’. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE. 2019, pp. 1–4.

[131] Yue Liu, Xin Wang, Yitian Yuan and Wenwu Zhu. ‘Cross-modal dual learning for sentence-to-video generation’. In: *Proceedings of the ACM International Conference on Multimedia*. ACM. 2019, pp. 1239–1247.

[132] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin and Baining Guo. ‘Swin Transformer: Hierarchical Vision Transformer using Shifted Windows’. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2021, pp. 10012–10022.

[133] Zhouyong Liu, Shun Luo, Wubin Li, Jingben Lu, Yufan Wu, Shilei Sun, Chunguo Li and Luxi Yang. 'Convtransformer: A convolutional transformer network for video frame synthesis'. In: *arXiv preprint arXiv:2011.10185* (2020).

[134] Ziwei Liu, Ping Luo, Xiaogang Wang and Xiaoou Tang. 'Deep learning face attributes in the wild'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2015, pp. 3730–3738.

[135] Ling Lo, Hong-Xia Xie, Hong-Han Shuai and Wen-Huang Cheng. 'MER-GCN: Micro-expression recognition based on relation modeling with graph convolutional networks'. In: *Proceedings of the IEEE conference on Multimedia Information Processing and Retrieval*. IEEE. 2020, pp. 79–84.

[136] Jonathan Long, Evan Shelhamer and Trevor Darrell. 'Fully convolutional networks for semantic segmentation'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2015, pp. 3431–3440.

[137] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll and Michael J Black. 'SMPL: A skinned multi-person linear model'. In: *ACM Transactions on Graphics* 34.6 (2015).

[138] Shitong Luo, Jiahan Li, Jiaqi Guan, Yufeng Su, Chaoran Cheng, Jian Peng and Jianzhu Ma. 'Equivariant Point Cloud Analysis via Learning Orientations for Message Passing'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2022, pp. 18932–18941.

[139] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou and Yue Zhang. 'An empirical study of catastrophic forgetting in large language models during continual fine-tuning'. In: *arXiv preprint arXiv:2308.08747* (2023).

[140] Yanni Ma, Yulan Guo, Hao Liu, Yinjie Lei and Gongjian Wen. 'Global Context Reasoning for Semantic Segmentation of 3D Point Clouds'. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. IEEE. 2020, pp. 2931–2940.

[141] Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich and Kevin Murphy. 'What's cookin'? interpreting cooking videos using text, speech and vision'. In: *Proceedings of the North American Chapter of the Association*

for Computational Linguistics: Human Language Technologies. ACL, 2015, pp. 143–152.

[142] Daniel Maturana and Sebastian Scherer. 'Voxnet: A 3d convolutional neural network for real-time object recognition'. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 922–928.

[143] Walied Merghani, Adrian K Davison and Moi Hoon Yap. 'A review on facial micro-expressions analysis: datasets, features and metrics'. In: *arXiv preprint arXiv:1805.02397* (2018).

[144] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes and Daniel Cohen-Or. 'Latent-NeRF for Shape-Guided Generation of 3D Shapes and Textures'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2023, pp. 12663–12673.

[145] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim and Rana Hanocka. 'Text2mesh: Text-driven neural stylization for meshes'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2022, pp. 13492–13502.

[146] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev and Josef Sivic. 'Howto100m: Learning a text-video embedding by watching hundred million narrated video clips'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2019, pp. 2630–2640.

[147] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi and Ren Ng. 'NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2020.

[148] Mehdi Mirza and Simon Osindero. 'Conditional generative adversarial nets'. In: *arXiv preprint arXiv:1411.1784* (2014).

[149] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze and Amit K Roy-Chowdhury. 'Learning joint embedding with multimodal cues for cross-modal video-text retrieval'. In: *Proceedings of the ACM on International Conference on Multimedia Retrieval*. ACM. 2018, pp. 19–27.

[150] Anish Mittal, Anush Krishna Moorthy and Alan Conrad Bovik. 'No-reference image quality assessment in the spatial domain'. In: *IEEE Transactions on Image Processing* 21.12 (2012), pp. 4695–4708.

[151] Gaurav Mittal, Tanya Marwah and Vineeth N Balasubramanian. 'Sync-draw: Automatic video generation using deep recurrent attentive architectures'. In: *Proceedings of the ACM International Conference on Multimedia*. ACM. 2017, pp. 1096–1104.

[152] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky and Tiberiu Popa. 'CLIP-Mesh: Generating Textured Meshes from Text Using Pretrained Image-Text Models'. In: *SIGGRAPH Asia Conference Papers*. ACM, 2022, pp. 1–8.

[153] Ali Mollahosseini, Behzad Hasani and Mohammad H Mahoor. 'Affectnet: A database for facial expression, valence, and arousal computing in the wild'. In: *IEEE Transactions on Affective Computing* 10.1 (2017), pp. 18–31.

[154] Thomas Müller, Alex Evans, Christoph Schied and Alexander Keller. 'Instant neural graphics primitives with a multiresolution hash encoding'. In: *ACM Transactions on Graphics* 41.4 (2022), pp. 1–15.

[155] Arsha Nagrani, Joon Son Chung and Andrew Zisserman. 'VoxCeleb: A Large-Scale Speaker Identification Dataset'. In: *Proceedings of the Interspeech*. 2017.

[156] Maureen O'sullivan, Mark G Frank, Carolyn M Hurley and Jaspreet Tiwana. 'Police lie detection accuracy: The effect of lie scenario.' In: *Law and Human Behavior* 33.6 (2009), p. 530.

[157] Augustus Odena, Christopher Olah and Jonathon Shlens. 'Conditional image synthesis with auxiliary classifier gans'. In: *Proceedings of the International Conference on Machine Learning*. PMLR. 2017, pp. 2642–2651.

[158] Itir Onal Ertugrul, László A Jeni and Jeffrey F Cohn. 'Facscaps: Pose-independent facial action coding with capsules'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. IEEE. 2018, pp. 2130–2139.

[159] Aäron van den Oord, Oriol Vinyals and Koray Kavukcuoglu. 'Neural discrete representation learning'. In: *Advances in Neural Information Processing Systems* (2017), pp. 6309–6318.

[160] Liang Pan, Zhongang Cai and Ziwei Liu. 'Robust partial-to-partial point cloud registration in a full range'. In: *arXiv preprint arXiv:2111.15606* (2021).

[161] Yingwei Pan, Zhaofan Qiu, Ting Yao, Houqiang Li and Tao Mei. 'To create what you tell: Generating videos from captions'. In: *Proceedings of the ACM International Conference on Multimedia*. ACM. 2017, pp. 1789–1798.

[162] Ben Poole, Ajay Jain, Jonathan T. Barron and Ben Mildenhall. 'DreamFusion: Text-to-3D using 2D Diffusion'. In: *Proceedings of the International Conference on Learning Representations*. 2023.

[163] Albert Pumarola, Jordi Sanchez-Riera, Gary Choi, Alberto Sanfeliu and Francesc Moreno-Noguer. '3dpeople: Modeling the geometry of dressed humans'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2019, pp. 2242–2251.

[164] Charles R Qi, Hao Su, Kaichun Mo and Leonidas J Guibas. 'Pointnet: Deep learning on point sets for 3d classification and segmentation'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 652–660.

[165] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan and Leonidas J Guibas. 'Volumetric and multi-view cnns for object classification on 3d data'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 5648–5656.

[166] Charles Ruizhongtai Qi, Li Yi, Hao Su and Leonidas J Guibas. 'Pointnet++: Deep hierarchical feature learning on point sets in a metric space'. In: *Advances in Neural Information Processing Systems* 30 (2017).

[167] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng and Kai Xu. 'Geometric Transformer for Fast and Robust Point Cloud Registration'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2022, pp. 11143–11152.

[168] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark et al. 'Learning

transferable visual models from natural language supervision'. In: *Proceedings of the International Conference on Machine Learning*. PMLR. 2021, pp. 8748–8763.

[169] Alec Radford, Luke Metz and Soumith Chintala. 'Unsupervised representation learning with deep convolutional generative adversarial networks'. In: *arXiv preprint arXiv:1511.06434* (2015).

[170] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya and Jon Shlens. 'Stand-alone self-attention in vision models'. In: *Advances in Neural Information Processing Systems* 32 (2019).

[171] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert and Sumit Chopra. 'Video (language) modeling: a baseline for generative models of natural videos'. In: *arXiv preprint arXiv:1412.6604* (2014).

[172] Yongming Rao, Jiwen Lu and Jie Zhou. 'Spherical fractal convolutional neural networks for point cloud recognition'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2019, pp. 452–460.

[173] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru and Brian Curless. 'FILM: Frame Interpolation for Large Motion'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2022, pp. 250–266.

[174] Renderpeople. https://renderpeople.com/. Accessed: 2023-07-21. 2021.

[175] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes and Daniel Cohen-Or. 'TEXTure: Text-Guided Texturing of 3D Shapes'. In: *ACM SIGGRAPH*. ACM, 2023.

[176] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser and Björn Ommer. 'High-resolution image synthesis with latent diffusion models'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2022, pp. 10684–10695.

[177] Olaf Ronneberger, Philipp Fischer and Thomas Brox. 'U-Net: Convolutional Networks for Biomedical Image Segmentation'. In: *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*. Ed. by Nassir Navab, Joachim Hornegger, William M. Wells and Alejandro F. Frangi. Springer, 2015, pp. 234–241.

[178] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies and Matthias Nießner. 'Faceforensics: A large-scale video dataset for forgery detection in human faces'. In: *arXiv preprint arXiv:1803.09179* (2018).

[179] Aurko Roy, Mohammad Saffar, Ashish Vaswani and David Grangier. 'Efficient Content-Based Sparse Attention with Routing Transformers'. In: *Transactions of the Association for Computational Linguistics* 9 (2021), pp. 53–68.

[180] Sara Sabour, Nicholas Frosst and Geoffrey E Hinton. 'Dynamic routing between capsules'. In: *Advances in Neural Information Processing Systems* 30 (2017).

[181] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa and Hao Li. 'Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization'. In: *Proceedings of the IEEE/CVF Conference on International Conference on Computer Vision*. IEEE. 2019, pp. 2304–2314.

[182] Shunsuke Saito, Tomas Simon, Jason Saragih and Hanbyul Joo. 'Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2020, pp. 84–93.

[183] Aditya Sanghi. 'Info3d: Representation learning on 3D objects using mutual information maximization and contrastive learning'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2020, pp. 626–642.

[184] Manolis Savva, Fisher Yu, Hao Su, Asako Kanezaki, Takahiko Furuya, Ryutarou Ohbuchi, Zhichao Zhou, Rui Yu, Song Bai, Xiang Bai et al. 'Large-scale 3D shape retrieval from ShapeNet Core55: SHREC'17 track'. In: *Proceedings of the Workshop on 3D Object Retrieval*. 2017, pp. 39–50.

[185] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev and Aran Komatsuzaki. 'Laion-400m: Open dataset of clip-filtered 400 million image-text pairs'. In: *arXiv preprint arXiv:2111.02114* (2021).

[186] John See, Moi Hoon Yap, Jingting Li, Xiaopeng Hong and Su-Jing Wang. 'Megc 2019–the second facial micro-expressions grand challenge'. In: *IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE. 2019, pp. 1–5.

[187] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh and Dhruv Batra. ‘Grad-cam: Visual explanations from deep networks via gradient-based localization’. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2017, pp. 618–626.

[188] Piyush Sharma, Nan Ding, Sebastian Goodman and Radu Soricut. ‘Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning’. In: *Proceedings of the Association for Computational Linguistics*. 2018, pp. 2556–2565.

[189] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu and Sanja Fidler. ‘Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis’. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 6087–6101.

[190] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta and Yaniv Taigman. ‘Make-A-Video: Text-to-Video Generation without Text-Video Data’. In: *Proceedings of the International Conference on Learning Representations*. 2023.

[191] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan and Surya Ganguli. ‘Deep unsupervised learning using nonequilibrium thermodynamics’. In: *Proceedings of the International Conference on Machine Learning*. PMLR. 2015, pp. 2256–2265.

[192] Riccardo Spezialetti, Federico Stella, Marlon Marcon, Luciano Silva, Samuele Salti and Luigi Di Stefano. ‘Learning to orient surfaces by self-supervised spherical cnns’. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 5381–5392.

[193] David Stap, Maurits Bleeker, Sarah Ibrahimi and Maartje ter Hoeve. ‘Conditional image generation and manipulation for user-specified content’. In: *arXiv preprint arXiv:2005.04909* (2020).

[194] Weiwei Sun, Andrea Tagliasacchi, Boyang Deng, Sara Sabour, Soroosh Yazdani, Geoffrey E Hinton and Kwang Moo Yi. ‘Canonical capsules: Self-supervised capsules in canonical pose’. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 24993–25005.

[195] Xiao Sun, Zhouhui Lian and Jianguo Xiao. 'SRINET: Learning strictly rotation-invariant representations for point cloud classification and segmentation'. In: *Proceedings of the ACM International Conference on Multimedia*. ACM. 2019, pp. 980–988.

[196] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens and Zbigniew Wojna. 'Rethinking the inception architecture for computer vision'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 2818–2826.

[197] Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao and Che Zheng. 'Synthesizer: Rethinking self-attention for transformer models'. In: *Proceedings of the International Conference on Machine Learning*. PMLR. 2021.

[198] Hugues Thomas. 'Rotation-Invariant Point Convolution With Multiple Equivariant Alignments'. In: *Proceedings of the International Conference on 3D Vision*. IEEE. 2020, pp. 504–513.

[199] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette and Leonidas J Guibas. 'Kpconv: Flexible and deformable convolution for point clouds'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2019, pp. 6411–6420.

[200] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit et al. 'Mlp-mixer: An all-mlp architecture for vision'. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 24261–24272.

[201] Federico Tombari, Samuele Salti and Luigi Di Stefano. 'Unique signatures of histograms for local surface description'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2010, pp. 356–369.

[202] 'Training deep networks for facial expression recognition with crowd-sourced label distribution'. In: *Proceedings of the ACM International Conference on Multimodal Interaction*. ACM. 2016, pp. 279–283.

[203] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani and Manohar Paluri. 'Learning spatiotemporal features with 3d convolutional networks'. In: *Proceedings of*

the IEEE/CVF International Conference on Computer Vision. IEEE, 2015, pp. 4489–4497.

[204] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski and Sylvain Gelly. 'Towards accurate generative models of video: A new metric & challenges'. In: *arXiv preprint arXiv:1812.01717* (2018).

[205] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen and Sai-Kit Yeung. 'Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2019, pp. 1588–1597.

[206] Jeya Maria Jose Valanarasu, Poojan Oza, Ilker Hacihaliloglu and Vishal M Patel. 'Medical Transformer: Gated Axial-Attention for Medical Image Segmentation'. In: *Proceedings of the Medical Image Computing and Computer Assisted Intervention*. Springer, 2021, pp. 36–46.

[207] Aaron Van den Oord, Yazhe Li and Oriol Vinyals. 'Representation learning with contrastive predictive coding'. In: *arXiv:1807.03748*. 2018.

[208] Laurens Van der Maaten and Geoffrey Hinton. 'Visualizing data using t-SNE'. In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605.

[209] Nguyen Van Quang, Jinhee Chun and Takeshi Tokuyama. 'CapsuleNet for micro-expression recognition'. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE. 2019, pp. 1–7.

[210] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser and Illia Polosukhin. 'Attention is all you need'. In: *Advances in Neural Information Processing Systems* 30 (2017).

[211] Eduardo Velloso, Andreas Bulling and Hans Gellersen. 'Motionma: motion modelling and analysis by demonstration'. In: *Proceedings of the Conference on Human Factors in Computing Systems*. ACM SIGCHI. 2013, pp. 1309–1318.

[212] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze and Dumitru Erhan. 'Phenaki: Variable Length Video Generation From Open Domain Textual

Description'. In: *Proceedings of the International Conference on Learning Representations*. 2023.

[213] Pascal Volino and N Magnenat Thalmann. 'Implementing fast cloth simulation with collision response'. In: *Proceedings of the Computer Graphics International*. IEEE. 2000, pp. 257–266.

[214] Chongyang Wang, Min Peng, Tao Bi and Tong Chen. 'Micro-attention for micro-expression recognition'. In: *Neurocomputing* (2020).

[215] Chongyang Wang, Min Peng, Tao Bi and Tong Chen. 'Micro-attention for micro-expression recognition'. In: *Neurocomputing* 410 (2020), pp. 354–362.

[216] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille and Liang-Chieh Chen. 'Axial-deeplab: Stand-alone axial-attention for panoptic segmentation'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2020, pp. 108–126.

[217] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang and Jie Shan. 'Graph attention convolution for point cloud semantic segmentation'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2019, pp. 10296–10305.

[218] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky and Raquel Urtasun. 'Deep parametric continuous convolutional neural networks'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2018, pp. 2589–2597.

[219] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang and William Yang Wang. 'Vatex: A large-scale, high-quality multilingual dataset for video-and-language research'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2019, pp. 4581–4591.

[220] Yiwei Wang, Lei Huang, Siwen Jiang, Yifei Wang, Jun Zou, Hongguang Fu and Shengyong Yang. 'Capsule networks showed excellent performance in the classification of hERG blockers/nonblockers'. In: *Frontiers in pharmacology* 10 (2020), p. 1631.

[221] Yue Wang and Justin M Solomon. 'Deep closest point: Learning representations for point cloud registration'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, 2019, pp. 3523–3532.

[222] Yue Wang and Justin M Solomon. 'Prnet: Self-supervised learning for partial-to-partial registration'. In: *Advances in Neural Information Processing Systems* 32 (2019).

[223] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein and Justin M Solomon. 'Dynamic graph cnn for learning on point clouds'. In: *ACM Transactions On Graphics* 38.5 (2019), pp. 1–12.

[224] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su and Jun Zhu. 'ProlificDreamer: High-Fidelity and Diverse Text-to-3D Generation with Variational Score Distillation'. In: *arXiv preprint arXiv:2305.16213* (2023).

[225] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Masayoshi Tomizuka, Kurt Keutzer and Peter Vajda. 'Visual Transformers: Token-based Image Representation and Processing for Computer Vision'. In: *CoRR* (2020).

[226] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro and Nan Duan. 'Godiva: Generating open-domain videos from natural descriptions'. In: *arXiv preprint arXiv:2104.14806* (2021).

[227] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang and Nan Duan. 'Nüwa: Visual synthesis pre-training for neural visual world creation'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2022, pp. 720–736.

[228] Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin and Michael Auli. 'Pay Less Attention with Lightweight and Dynamic Convolutions'. In: *Proceedings of the International Conference on Learning Representations*. 2019.

[229] Tianyi Wu, Yu Lu, Yu Zhu, Chuang Zhang, Ming Wu, Zhanyu Ma and Guodong Guo. 'GINet: Graph interaction network for scene parsing'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2020, pp. 34–51.

[230] Wayne Wu, Yunxuan Zhang, Cheng Li, Chen Qian and Chen Change Loy. 'Reenactgan: Learning to reenact faces via boundary transfer'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2018, pp. 603–619.

[231] Wenxuan Wu, Zhongang Qi and Li Fuxin. 'Pointconv: Deep convolutional networks on 3d point clouds'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 9621–9630.

[232] Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin and Song Han. 'Lite transformer with long-short range attention'. In: *Proceedings of the International Conference on Learning Representations*. 2020.

[233] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang and Jianxiong Xiao. '3d shapenets: A deep representation for volumetric shapes'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2015, pp. 1912–1920.

[234] Weihao Xia, Yujiu Yang, Jing-Hao Xue and Baoyuan Wu. 'Tedigan: Text-guided diverse face image generation and manipulation'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2021, pp. 2256–2265.

[235] Hong-Xia Xie, Ling Lo, Hong-Han Shuai and Wen-Huang Cheng. 'An overview of facial micro-expression analysis: Data, methodology and challenge'. In: *IEEE Transactions on Affective Computing* (2022).

[236] Hong-Xia Xie, Ling Lo, Hong-Han Shuai and Wen-Huang Cheng. 'Au-assisted graph attention convolutional network for micro-expression recognition'. In: *Proceedings of the ACM International Conference on Multimedia*. ACM. 2020, pp. 2871–2880.

[237] Jin Xie, Guoxian Dai, Fan Zhu, Edward K Wong and Yi Fang. 'Deepshape: Deep-learned shape descriptor for 3d shape retrieval'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 39.7 (2016), pp. 1335–1345.

[238] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas and Or Litany. 'Pointcontrast: Unsupervised pre-training for 3d point cloud understanding'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2020, pp. 574–591.

[239] Jianyun Xu, Xin Tang, Yushi Zhu, Jie Sun and Shiliang Pu. 'SGMNet: Learning Rotation-Invariant Point Cloud Representations via Sorted Gram Matrix'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2021, pp. 10468–10477.

[240] Jun Xu, Tao Mei, Ting Yao and Yong Rui. 'Msr-vtt: A large video description dataset for bridging video and language'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2016, pp. 5288–5296.

[241] Mingye Xu, Zhipeng Zhou and Yu Qiao. 'Geometry sharing network for 3d point cloud classification and segmentation'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. AAAI Press, 2020, pp. 12500–12507.

[242] Mutian Xu, Runyu Ding, Hengshuang Zhao and Xiaojuan Qi. 'PAConv: Position Adaptive Convolution with Dynamic Kernel Assembling on Point Clouds'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2021, pp. 3173–3182.

[243] Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang and Ulrich Neumann. 'Grid-gcn for fast and scalable point cloud learning'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2020, pp. 5661–5670.

[244] Xiangyu Xu, Hao Chen, Francesc Moreno-Noguer, Laszlo A Jeni and Fernando De la Torre. '3D human pose, shape and texture from low-resolution images and videos'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9 (2021), pp. 4490–4504.

[245] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng and Yu Qiao. 'Spidercnn: Deep learning on point sets with parameterized convolutional filters'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2018, pp. 87–102.

[246] Wen-Jing Yan, Xiaobai Li, Su-Jing Wang, Guoying Zhao, Yong-Jin Liu, Yu-Hsin Chen and Xiaolan Fu. 'CASME II: An improved spontaneous micro-expression database and the baseline evaluation'. In: *PloS one* 9.1 (2014).

[247] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang and Shuguang Cui. 'Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 5589–5598.

[248] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou and Qi Tian. 'Modeling point clouds with self-attention and gumbel subset sampling'.

In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2019, pp. 3323–3332.

[249] Xi Yang, Ding Xia, Taichi Kin and Takeo Igarashi. ‘Intra: 3d intracranial aneurysm dataset for deep learning’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 2656–2666.

[250] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov and Quoc V Le. ‘Xlnet: Generalized autoregressive pretraining for language understanding’. In: *Advances in Neural Information Processing Systems* 32 (2019).

[251] Zi Jian Yew and Gim Hee Lee. ‘Rpm-net: Robust point matching using learned features’. In: *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*. 2020, pp. 11824–11833.

[252] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer and Leonidas Guibas. ‘A scalable active framework for region annotation in 3d shape collections’. In: *ACM Transactions on Graphics* 35.6 (2016), pp. 1–12.

[253] Yang You, Yujing Lou, Qi Liu, Yu-Wing Tai, Lizhuang Ma, Cewu Lu and Weiming Wang. ‘Pointwise rotation-invariant network with adaptive sampling and 3d spherical voxel convolution’. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. AAAI Press, 2020, pp. 12717–12724.

[254] Gilbert Youmans. ‘Measuring lexical style and competence: The type-token vocabulary curve’. In: *Style* (1990), pp. 584–599.

[255] Jianhui Yu, Chaoyi Zhang, Heng Wang, Dingxin Zhang, Yang Song, Tiange Xiang, Dongnan Liu and Weidong Cai. ‘3d medical point transformer: Introducing convolution to attention networks for medical point cloud analysis’. In: *arXiv preprint arXiv:2112.04863* (2021).

[256] Ruixuan Yu, Xin Wei, Federico Tombari and Jian Sun. ‘Deep positional and relational feature learning for rotation-invariant point cloud analysis’. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2020, pp. 217–233.

[257] Wentao Yuan, Benjamin Eckart, Kihwan Kim, Varun Jampani, Dieter Fox and Jan Kautz. 'Deepgmr: Learning latent gaussian mixture models for registration'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2020, pp. 733–750.

[258] Yifei Zeng, Yuanxun Lu, Xinya Ji, Yao Yao, Hao Zhu and Xun Cao. 'AvatarBooth: High-Quality and Customizable 3D Human Avatar Generation'. In: *arXiv preprint arXiv:2306.09864* (2023).

[259] Bowen Zhang, Hexiang Hu and Fei Sha. 'Cross-modal and hierarchical modeling of video and text'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2018, pp. 374–390.

[260] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao and Hongsheng Li. 'Pointclip: Point cloud understanding by clip'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 8552–8562.

[261] Songyang Zhang, Xuming He and Shipeng Yan. 'Latentgnn: Learning efficient non-local relations for visual recognition'. In: *Proceedings of the International Conference on Machine Learning*. PMLR. 2019.

[262] Zhiyuan Zhang, Binh-Son Hua, Wei Chen, Yibin Tian and Sai-Kit Yeung. 'Global context aware convolutions for 3d point cloud understanding'. In: *International Conference on 3D Vision*. IEEE. 2020, pp. 210–219.

[263] Zhiyuan Zhang, Binh-Son Hua, David W Rosen and Sai-Kit Yeung. 'Rotation invariant convolutions for 3D point clouds deep learning'. In: *International Conference on 3D Vision*. IEEE. 2019, pp. 204–213.

[264] Chen Zhao, Jiaqi Yang, Xin Xiong, Angfan Zhu, Zhiguo Cao and Xin Li. 'Rotation invariant point cloud analysis: Where local geometry meets global topology'. In: *Pattern Recognition* 127 (2022), p. 108626.

[265] Guoying Zhao and Matti Pietikainen. 'Dynamic texture recognition using local binary patterns with an application to facial expressions'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.6 (2007), pp. 915–928.

[266] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr and Vladlen Koltun. 'Point transformer'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2021, pp. 16259–16268.

[267] Haoran Zhou, Yidan Feng, Mingsheng Fang, Mingqiang Wei, Jing Qin and Tong Lu. 'Adaptive Graph Convolution for Point Cloud Analysis'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2021, pp. 4965–4974.

[268] Ling Zhou, Qirong Mao and Luoyang Xue. 'Dual-inception network for cross-database micro-expression recognition'. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*. 2019, pp. 1–5.

[269] Luowei Zhou, Chenliang Xu and Jason J Corso. 'Towards automatic learning of procedures from web instructional videos'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. AAAI Press, 2018.

[270] Hao Zhu, Wayne Wu, Wentao Zhu, Liming Jiang, Siwei Tang, Li Zhang, Ziwei Liu and Chen Change Loy. 'CelebV-HQ: A Large-Scale Video Facial Attributes Dataset'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2022, pp. 650–667.

[271] Luyang Zhu, Konstantinos Rematas, Brian Curless, Steven M Seitz and Ira Kemelmacher-Shlizerman. 'Reconstructing NBA players'. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2020, pp. 177–194.

[272] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev and Josef Sivic. 'Cross-task weakly supervised learning from instructional videos'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2019, pp. 3537–3545.

Appendix

Abbreviations

SoTA: State-of-The-Art

3D: Three-dimensional

AU: Action Unit

NLP: Natural Language Processing

FPS: Farthest Point Sampling

KNN: K Nearest Neighbors

GCN: Graph Convolutional Network

MGR: Multi-Graph Reasoning

PCR: Point Cloud Registration

AIT: Aligned Integration Transformer

PCA: Principal Component Analysis

ICP: Iterative Closest Point

EVD: Eigenvalue Decomposition

SGD: Stochastic Gradient Descent

mIoU: mean Intersection over Union

mAP: mean Average Precision

LRF: Local Reference Frame

GRF: Global Reference Frame

SMPL: Skinned Multi-Person Linear Model

NeRF: Neural Radiance Field

LBS: Linear Blend Skinning

SDS: Score Distillation Sampling

DSD: Denoising Score Distillation

PBR: Physically-Based Rendering

BRDF: Bidirectional Reflectance Distribution Function

CFG: Classifier-Free Guidance

SH: Spherical Harmonic

ME: Micro-Expression

MER: Micro-Expression Recognition

MES: Micro-Expression Synthesis FACS: Facial Action Coding System

METT: Micro Expression Training Tool

SETT: Subtle Expression Training Tool

LBP: Local Binary Pattern

LQP: Local Quantized Pattern

LBP-TOP: Local Binary Pattern with Three Orthogonal Planes

CNN: Convolutional Neural Network

GAN: Generative Adversarial Network

DCGAN: Deep Convolutional Generative Adversarial Network

WGAN: Wasserstein Generative Adversarial Network

CGAN: Conditional Generative Adversarial Network

ACGAN: Auxiliary Classifier Generative Adversarial Network

GRM: Graph Reasoning Module

LOSO: Leave-One-Subject-Out

CDE: Cross-Database Evaluation

UF1: Unweighted F1-score

UAR: Unweighted Average Recall

SDE: Single Database Evaluation

SE: Squeeze-and-Excitation

PCFG: Probabilistic Context-Free Grammar

POS: Part-Of-Speech

FID: Frechet Image Distance

FVD: Frechet Video Distance

MdR: Median Rank

MnR: Mean Rank