SHAPE-ASSISTED MULTIMODAL PERSON RE-IDENTIFICATION

by

Haidong Zhu

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(Department of Computer Science)

May 2024

Dedication

To my parents, Weicheng Zhu and Maomei Luo, for their unconditional support and love.

To all my friend and all the people who support my life.

Acknowledgements

The past five years at the University of Southern California have been undoubtedly one of the most precious experiences of my life, not only because of the unique and life-changing black-swan events but also due to the talented and helpful individuals I've had the fortune to encounter, to whom I will always be grateful. Five years ago, I came across a recruitment advertisement on Zhihu, posted by Jiyang, and submitted my resume. When Zhenheng and Prof. Ram Nevatia contacted me in February, I knew it was time to make my commitment. Reflecting on the past five unprecedented years, I am forever thankful for the decision I made and appreciative of the people I have worked with during this time.

First and foremost, I would like to express my sincere gratitude to my advisor, Prof. Ram Nevatia. When I joined the lab, most of the senior students had just graduated, which made my first two years challenging as I struggled to understand what it takes to be a qualified PhD student. Prof. Nevatia provided guidance not only on academic research but also in my personal life. Academically, Prof. Nevatia helped filter my ideas, providing insight into what is considered a correct direction, as well as offering instructions and feedback for various ablations and analyses. In life, Prof. Nevatia also offered guidance and support. I am equally grateful to Prof. Stefanos Nikolaidis and Prof. Mohammad Soleymani for being part of my qualification committee, and to Prof. Antonio Ortega and Prof. Ulrich Neumann for being on my qualification and dissertation committee, all of whom provided insightful suggestions and support. Additionally, I would like to express my gratitude to Prof. Yueqi Duan and Prof. Donglai Wei, who introduced me to

modern computer vision tasks and supported me throughout the connection of my undergraduate and PhD studies.

Furthermore, I wish to extend my appreciation to all the mentors and colleagues I've had the privilege to collaborate with. A special thanks goes to Zhaoheng Zheng for our fruitful collaborations and discussions. My gratitude also extends to Jiyang Gao and Zhenheng Yang for introducing me to the IRIS lab, as well as to Xuefeng Hu, Arka Sadhu, and Changcheng Fu, with whom I've had the fortune to work as labmates during my PhD journey. I am also grateful to my mentors and managers with whom I've been lucky to work, including Yuan Ye, Jiajia Luo, Cheng-hao Kuo, and most notably, Tianyu Ding and Luming Liang. Your guidance and support have provided me with cherished memories of three unforgettable summers and have prepared me for a career as a qualified researcher and for life ahead. I also want to acknowledge all the authors I've had the pleasure of working with, including Tianyi Chen, Yiqi Zhong, Jinxin Zhou, Guangzhi Wang, Wanrong Zheng, Pranav Budhwant, Kan Chen, Yiheng Zhu, Chuanzi He, Ilya Zharkov, Prof. He Wang, Prof. Li Yi, Prof. Leonidas Guibas, Chi Liu, Lu Xia, and many others.

Last but not least, my sincere thanks go to my parents, Weicheng Zhu and Maomei Luo, for their unconditional support and love throughout my life, and to my friends for their unwavering support, especially during the unprecedented times.

Table of Contents

List of Tables

List of Figures

Abstract

Finding the identity of a person from a non-overlapping input image or video, known as person re-identification, is a classic and important task in biometric understanding. Identifying the corresponding identity requires extracting the representations of the person and distinguishing them across different individuals, where the representation can be human appearance, specific walking patterns, and body shape. Each representation can be understood as a specific modality and has its own strengths and weaknesses, while different modalities can sometimes complement each other. Therefore, combining two or more modalities introduces a more robust system for person re-identification.

In this thesis, we cover different combinations of biometric representations for whole-body person re-identification, including appearance, gait, and body shape. Appearance is one of the most widely used biometric signals as it provides abundant information. Gait, represented as skeleton or binary silhouette sequences, captures the walking patterns of a person. Different from other representations, 3-D shape complements the body information with external human body shape prior and enhances the appearance captured in the 2-D images. Body shape also provides a strong prior of the person and helps complete the body shape to deal with occlusions. We discuss the combination of different representations and biometric signals that leverage their strengths, along with a system using the three signals for person re-identification in the wild. As the current body shapes used for person re-identification are usually not accurate enough to provide a distinguishable signal, we further discuss the improvement of the representations and how they can be applied for downstream vision tasks, such as person identification.

We begin with three works that explicitly extract and combine different modalities for re-identification, including two gait representations (silhouettes and skeletons), two different shape-related modalities (gait and 3-D body shape), and the additional use of appearance along with the two shape-related modalities. Although 3-D body shape offers invaluable external shape-related information that 2-D images lack, existing body shape representations often fall short in accuracy or demand extensive image data, which is unavailable for re-identification tasks. Following this, we explore the potential of using more accurate body shape to further improve the model and introduce two other methods for more accurate 3-D shape representation and reconstruction: Implicit Functions (IF) and Neural Radiance Fields (NeRF). Since a fine-grained representation is needed for downstream vision tasks, we discuss how to include more semantic representation to assist the training of the 3-D reconstruction model and how it can aid with a limited number of input views. Lastly, with the fine-grained representation, we discuss using them for body shape representation to enhance appearance for person re-identification. We conclude the thesis with potential future work for further improvements.

Chapter 1

Introduction

1.1 Background and Motivation

Person re-identification [239, 97, 132], which recognizes the identity of a person from non-overlapping cameras, is a classic task in computer vision and crucial for daily security. Given a single image or a sequence of images as a video containing the tracklets of the person, identifying the person involves matching the person recorded with the identities in a gallery of known identities, where the image or video used for identification is termed as the query image or video. Depending on the circumstances, such identification has different requirements. When verifying a person's identity [101, 221], identifying the person requires a specific matching score with the given gallery identity and needs to ensure the correct matching goes over the threshold while the incorrect matching stays below the threshold. In other circumstances, such as recognizing the person or retrieving the identity of the person from a tracklet [240, 207, 72, 83, 264, 132], a feasible person identification system needs to find the top-match results from a gallery with millions of candidate IDs. Different tasks require different perspectives and have different requirements.

In this thesis, we focus on retrieving the identity from a large gallery with the query image or video. We focus on whole-body input, since face [30, 88, 36] or fingerprint images [40, 56, 54] are difficult to capture in the open environment. Based on the type of input, a system can recognize the person based on different modalities of the whole-body images, such as appearance [207, 200], body shape [111, 130, 233, 198, 15] and

Modality	Standing Videos	Different Clothing	Turbulence & Occlusion
Gait		✓	✓̸
Body shape	✓̸	✓̸	✓
Appearance	✓	✓̸	✓̸

Figure 1.1: Circumstances where different modalities can be applied to different cases. A ✓ represents that the corresponding modality can handle such a case with reasonable performance, and a ✓̸ indicates that the modality can handle the case but with limited performance. Gait is unreliable in stationary videos, and appearance changes when subjects wear different clothing. The imprecise reconstruction of 3-D body shapes leads to unstable predictions, while the human prior assists in cases of occlusion.

gait [12, 44, 106, 237, 256, 254], which we present the circumstance they can handle in Figure 1.1. To extract identity specific information from a single image, appearance [240, 207, 72, 83, 264, 132] is the most widely used modality for re-identification since it includes the richest information needed to recognize a person. However, it may suffer in clothes changing cases for long-term re-identification [59, 27]. Body shapes [111, 130, 233, 198, 15] can provide a complete body reconstruction with external knowledge, which is helpful for occlusion or clothes changing cases. Although being useful as an additional distinguishable information, the reconstructed body shape usually lacks accuracy [255] and cannot be used for direct identification.

Compared with single-frame images, videos [74, 59, 200] include frame sequences capturing specific activities. This makes identification with specific patterns, such as gait, possible. Gait [226, 179] describes the pattern of the person's walking sequence and is one of the most widely used modalities in video for recognition. Compared with other modalities, gait has the advantage of being observable from a long distance and without the subject's cooperation. As a higher-level information, gait can be described with

different representations, such as skeletons [180] or binarized silhouettes [44, 106], which describe the joint positions and boundary of human segmentation respectively. Skeletons [257] can represent the movement of joints without the interference of the body shape or carried-on objects, but their detection is often noisy [254, 242], and error in joint detection can significantly degrade performance. Silhouettes, however, are more robust since pixel-level segmentation errors do not have a severe impact on the final silhouettes. However, such representation always includes the carried-on objects and clothes [226], which change the body contour from case to case.

As different input modalities have their own strengths and weaknesses [255], We propose using a combination of these diverse modalities. Instead of depending solely on a single modality for person identification, we implement various levels of fusion to construct a more robust re-identification system. Specifically, 3-D shape augments body information with external human body shape priors, enriching the appearance data captured in 2-D images. While body shape contributes invaluable external shape-related information absent in 2-D images, current body shape representations [119, 139] often lack precision or require extensive image data, typically not available for re-identification tasks. This work focuses on various biometric representations for a holistic approach to whole-body person re-identification, with a special focus on the application of body shape.

1.2 Challenges

To collect and analyze shape-based multiple modalities for person re-identification, one must address the following major challenges:

- **Multimodal selection and analysis.** Prior to applying 3-D human body shape for person re-identification, an in-depth analysis of the combinations of different modalities and the requirements for the final system is necessary, as depicted in Figure 1.1.

- **Improvement of 3-D body shape accuracy.** Despite the additional information provided by 3-D representations beyond 2-D features, their generalizability and semantic interpretation present two significant challenges that need resolution before their application in re-identification.

- **Use of body shape for person re-identification.** Determining how to integrate shape with appearance is crucial when utilizing 3-D representations, as it provides additional guidance for distinguishing individuals.

This thesis focuses on the multimodalities utilized in person re-identification. We commence by merging different modalities in multimodal person re-identification and evaluating each one's contribution. Subsequently, we concentrate on employing body shape for person re-identification, specifically focusing on the improvement of 3-D reconstruction and representation, and then discuss how to utilize refined body shapes for person re-identification.

1.3 Shape-Assisted Multimodal Person Re-Identification

In a re-identification model, the person can be recognized based on their appearance (RGB images) [240, 207, 72, 83, 264, 132], body shape (3-D representation) [111, 130, 233, 198, 15], and gait representations (silhouette and skeletons [257]) [256, 254, 12, 44, 106], as illustrated in Figure 1.1. Among these modalities, silhouettes and 3-D representation capture the overall body shape [253] for 2-D and 3-D representation, respectively. Compared with other modalities, body shape provides extra knowledge with a human body prior [119, 139], enriching the representation for re-identification in addition to other modalities.

For a shape-assisted multimodal recognition model, we can use the body shape in two ways: 1) using it as the explicit model input to provide re-identification information [111, 15], or 2) use it as the implicit supervision to enhance other modalities [249]. When using the body shape explicitly as the model input,

the re-identification model can rely on different modalities that describe the same patterns, such as skeletons and silhouettes, or different patterns across different inputs, such as body shape with appearance. Different levels of fusion are able to provide different levels of understanding using body shapes, which we discuss separately in this thesis. In addition, we discuss the different ways of combining body shape for multimodal person identification, including explicit combination and implicit supervision.

1.3.1 GaitSTR: Multimodal Re-Identification for Gait Recognition

Gait can be represented as both silhouettes [226] and skeletons [180]. As skeleton detection is less reliable due to the consequences of incorrect joint predictions, using binarized silhouettes, which provide the 2-D human body segmentations as the shape prior, offers external shape-related information to overcome the joint errors in predictions for gait recognition. In this part, body shape is used as a complementary input to assist the existing model for more robust recognition. We focus on the low-level fusion between different representations of the same modality, gait, and use silhouette to refine the skeletons for person re-identification based on gait recognition. Since skeletons and silhouettes are both the representation of the gait, a same sequence of two representations should provide consistent gait output. Motivated by this, we introduce GaitSTR, which use sequential two-stream refinement between silhouettes and skeletons for gait recognition. Focusing on the same sequence, we fuse the information of the skeleton and silhoeutte branch and correct the jitters in noisy skeletons. We evaluate this on four public datasets, including CASIA-B, OU-MVLP, GREW and Gait3D, and show state-of-the-art performance compared with other single or multi-modal gait recognition methods.

1.3.2 GaitHBS: Distilling 3-D Body Shape for Gait Recognition

As the identity across different frames in the same gait video does not change, the body shape extracted for each frame should also be consistent. Motivated by this, we introduce GaitHBS, using Human Body Shape

for gait recognition, and discuss distilling 3-D body shape from silhouette sequences to assist activity-based gait recognition for an extra shape guidance and enhancement. Distilling body shape across different frames also provide additional regulation for ensuring the shapes across different frames are consistent. We assess the model on CASIA-B and OU-MVLP using different state-of-the-art models as backbone, and show improvement for average performance and reduce the standard deviation for results across different camera views with the assist of distilled body shapes.

1.3.3 ShARc: Multimodal Re-Identification with Different Representations

Instead of relying on one of the representations for person re-identification, ShARc, as shown in Figure 1.2, focuses on directly using body shape representations as the input for re-identification, along with other modalities [255], such as appearance and gait. ShARc represents using **Sh**ape and **A**ppearance for **Re**cognition. As body shape provides external knowledge for occlusion cases and a clear body prior for images or frames with severe degradation, such fusion across different modalities is able to combine the different representations together for a more robust recognition. Different from other existing works, ShARc explicitly analyzes the performance of each modality and encode shape and pose along with appearance separately in the system. By separating the two representations, we also assess their contribution to the final pipeline. We tested ShARc on three public datasets with clothes changes for long-term person re-identification, including BRIAR, MEVID and CCVID, and present superior performance compared with other state-of-the-art methods.

1.4 3-D Representation and Reconstruction for Re-Identification

Since body shape acts as an important addition to the 2-D representation, reconstructing 3-D body shapes and extracting their representations are crucial for multimodal person re-identification tasks. Although 3-D representation shows good potential for providing guidance for recognizing the identity of people,

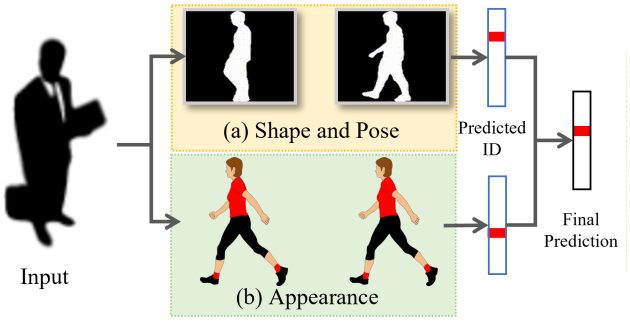


Figure 1.2: An overall illustration of the ShARC pipeline. ShARC recognizes the identity of the person based on two different branches: pose and shape, along with appearance separately. Two scores are aggregated together with a weighted average for the final prediction.

the quality of the 3-D representation still limits the overall performance of the model. Existing 3-D representations can be divided into two different branches. Earlier representations, such as point clouds [143, 145, 201, 37, 1, 148] and meshes [57, 61, 191, 168], provide sufficient generalizability across different body shapes with a strong prior, but they lack good distinguishability across different models. Recent methods include implicit representations, such as NeRF [129, 128] and Implicit Functions [8, 161, 186, 185, 135], for more precise modeling. Although the reconstructed shapes are much more fine-grained than the earlier methods, they either lack an image-level representation or require optimization for each scene, which makes them not feasible to be used in downstream tasks.

Based on the more recent representations, we discuss including semantic representations for reconstruction and rendering [250] to extract the scene-level representation for downstream vision tasks, such as person re-identification. Such representation provides a semantic understanding of what the scene includes, which can also help improve the quality of the reconstructed shape in reverse. We discuss two different ways for refinement in reconstruction with object or scene-level representations in this thesis, from scratch to fine-grained construction [38] and from multiview to single-view generalizable results [250].

1.4.1 Curriculum DeepSDF: Semantic Assistance for Object-level Construction

Instead of training the fine-grained representation from the initial step, we enhance the reconstruction quality progressively [39, 155] with curriculum learning. The process begins with instructing the semantic representations to describe the basic shape, gradually becoming more precise and fine-grained. This step-by-step training approach allows the scene representation to capture different levels of features and aids in understanding what each level of shape represents, preventing the model from missing fine-grained details due to difficulties in comprehending the shape at early stages. We build this approach on DeepSDF [136] as Curriculum DeepSDF, and achieve significant better reconstruction results on ShapeNet [10] dataset, even with exactly the same network architecture during inference.

1.4.2 CaesarNeRF: Scene-level Representation for Generalizable Rendering

Due to the pixel-level rendering scheme in Neural Rendering [129, 128], a NeRF model typically requires fine-tuning on a new scene or numerous reference views for generalizable rendering. We introduce CaesarNeRF, using scene-level representation to help each pixel understand its relationship with others. Furthermore, to address the lack of camera position information in scene-level representation, we integrate calibration to resolve inconsistencies across different frames. This approach enables the extraction of semantic representation from a fine-grained rendering model and its application in downstream tasks. We have accessed the model on four public datasets, including LLFF [127], Shiny [208], mip-NeRF 360 [4] and MVImgNet [227], and show better rendering results on generalizable rendering settings, especially with limited reference views.

1.4.3 SEAS: Implicit Shape Representation and Shape as Supervision

With improved body shape representation as implicit function, we aim to enhance other modalities with body shape. Compared with the previous two methods of explicitly using body shape for re-identification,

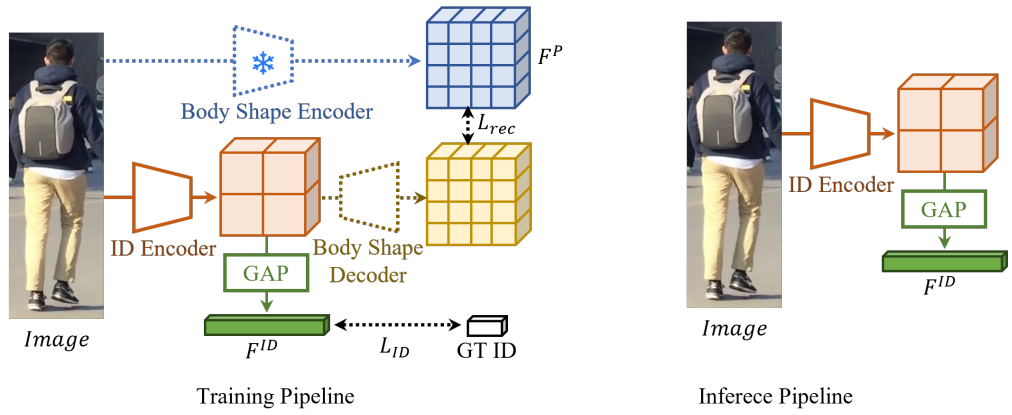


Figure 1.3: With a body shape encoder to provide body shape feature map, SEAS use 3-D body shape as supervision during training by decoding the body shape from the identity feature map with a trainable body shape decoder, while the 3-D branch is dropped during inference. More details are available in Chapter 8.

implicit representation [136, 38] provides a coherent understanding between different modalities and can fuse the information across different modalities in a more subtle approach. Considering that implicit representations are not well developed, we cover the improvement of the representation itself, from the shape-only representation to the shape with appearance ones, and then using it as implicit supervision for other modalities that used for person re-identification. We name our method as SEAS, abbreviated from ShapE-Aligned Supervision, as shown in Figure 1.3. SEAS only uses 3-D body shape for supervision during training and does not include the body inference branch during inference. Experiments on both video-based and frame-based person re-identification datasets show that using body shape as supervision is able to provide extra guidance compared to other combinations.

1.5 Outline and Contributions

For the remainder of this thesis, we first discuss the related literature for different person identification methods and shape representation and reconstruction in Chapter 2. We then delve into using body shape for multimodal person re-identification in our main section. We first discuss using body shape as explicit input for gait recognition and person re-identification tasks in Chapter 3, Chapter 4, and Chapter 5. This

includes using body shape as guidance for skeleton correction in gait recognition as outlined in Chapter 3, using body shape as the explicit input and guidance for gait recognition in addition to silhouettes as described in Chapter 4, and explicitly combining body shape representation with appearance and skeletons for person re-identification in Chapter 5.

Following the explicit combination of body shape for re-identification, we discuss the improvement of body shape representation with implicit representations and their use in person re-identification in Chapter 6, Chapter 7, and Chapter 8. In Chapter 6, we explore how to enhance the shape-based implicit function with a curriculum training scheme without increasing inference complexity. In Chapter 7, we discuss using semantic representations for neural radiance fields to provide possible solutions for generalizable few-shot rendering cases. Finally, with state-of-the-art implicit representation, we discuss using shape as guidance rather than input for person re-identification in Chapter 8.

With these two approaches to utilizing shape in person-identification, we conclude in Chapter 9 and discuss potential future directions for shape-assisted person re-identification.

The main contributions of this thesis are as follows:

- We theoretically discuss two different methods of using body shape for person re-identification, including explicit combination with other modalities [255] and supervision [249] for the extraction of identity features based on other modalities.

- For the combination of body shape with other modalities for multimodal person re-identification that use body shape as input, we explore three different levels of body shape fusion. These include using body shape to correct skeletons for gait recognition [242, 254], employing it as a 3-D body shape representation input for gait recognition [256], and using it as a shape representation in addition to appearance-based recognition for person re-identification [255].

- Regarding implicit supervision with shape representation, we first discuss improvements in shape reconstruction, including the enhancement of a shape-based implicit function using curriculum learning [38] and the use of calibrated semantic representation for scene-level understanding in generalizable neural radiance fields [250]. With fine-grained shape representation, we explore using it as supervision for an appearance-based model, rather than as a direct input, to foster a coherent understanding across different modalities [249].

Chapter 2

Related Work

In this chapter, we discuss the latest progress in related works, focusing on two main topics: person re-identification and the development of shape reconstruction techniques. For person re-identification, we delve into the progress made in recognizing a person's identity using various modalities, including appearance, gait, and multimodal recognition. Regarding shape reconstruction, we explore the advancements in template-based body shape reconstruction and other recent developments in implicit shape reconstruction.

2.1 Appearance-based Person Re-Identification

Recognizing people's identities based on their whole-body appearance focuses on matching individuals across different camera views or circumstances [239, 97, 132]. Compared to other modalities, such as fingerprints [40, 56, 54], palms [163, 236, 245, 53, 63, 81], or faces [30, 88, 36], re-identification using whole-body images [240, 207, 72, 83, 264] enables capturing visual representations from long distances and can maximally include all possible patterns for identifying a person. Given the tracklet of the person, raw images of the whole body are the most widely used among all modalities, since they contain all patterns related to the person and do not require further processing.

To recognize a person based on whole-body recognition, the visual input to the model can be either single-frame images [199, 209, 11, 181, 241, 84] or videos [74, 59, 200]. Earlier methods focused on short-term person re-identification, where the clothes across different cameras of the same person did not change, and they relied on single-frame images instead of videos due to computation restrictions. Researchers developed various methods to extract information for minimizing the distances between correct matches while maximizing the distances between different identities, including part-based [90, 260, 169, 22, 70, 100, 50] and attention-based methods [200, 17, 165, 234].

In addition to re-identification with single-image input, recognizing a person with video input is discussed using a sequence of frames to provide more reliable results for handling bad camera views or degraded images with more robust predictions. To combine different frames in a sequence, existing methods focus on temporal pooling [96, 112, 238, 49] or recurrent networks [26, 123, 248] to convert a video into a single feature representation. Recently, researchers have been focusing on attention-based mechanisms [74, 200, 167, 172, 218, 47, 110, 95] to combine different frames both spatially and temporally. Compared to direct pooling or recurrent networks, temporal and spatial attention helps the network focus on the most important parts for person recognition.

Apart from recognizing a person with the same clothes of the same identity in short-term videos, researchers are also focusing on long-term re-identification [59, 27], where the clothes of the same identities in the queried examples and galleries are not the same. This indicates that relying on clothing for recognition is no longer reliable. To handle this problem, existing methods focus on separating clothes patterns from the body for re-identification. However, as clothes cover most of the body, ignoring them can lead to the loss of the majority of useful information, making re-identification based on single-modal input less reliable when not having them as distinguishable information.

2.2 Gait Recognition

Different from appearance-based person re-identification, gait recognition focuses on identifying a person based on their walking patterns. Compared with appearance-based recognition methods, gait patterns, usually captured via binarized silhouettes [226, 179] describing body shape contours, reduce the negative impact of clothing changes for identification but introduce different appearance variations with body contours. Due to the lack of RGB patterns, it is challenging to infer body information directly from silhouettes. To address this, some researchers [44, 106] focus on part-based recognition, while others [76, 12, 42] extract framewise consistencies for identification.

Due to the limited information in silhouettes, recent research [3, 256, 180, 162, 62] focuses on external modalities to assist silhouettes for identification. GaitGraph [180], GaitMix [254] and GaitRef [254] apply or refine HRNet [190] for joint detection and uses the generated pose sequence for identification. Gait3D [237], GaitHBS [256], and ModelGait [99] focus on extracting or using body shapes alongside silhouettes for gait recognition, intending to provide more information for part separation. LiDARGait [162] employs point clouds instead of silhouettes for body shape description. Some researchers [103, 62] also integrate RGB images with silhouettes for gait understanding. Since these methods still focus on gait representation, they can only apply to walking sequences for identification.

2.3 Multimodal Person Re-Identification

Instead of simply relying on a single modality to capture the identity of a person, researchers are introducing additional modalities for the recognition of individuals, which serve as complementary information when appearance lacks sufficient distinguishable features. As the most commonly used modalities for identifying persons are their appearance and gait, which are both restricted to the 2-D framewise domain, the most commonly used extra modality to recognize the person is the 3-D body shape [111, 130, 233, 198, 15]

with depth awareness in addition to the 2-D representation. For 3-D body modeling, existing methods [111, 60] employ shape priors from SMPL [119]. However, these methods don't accurately model the body shape and encode the body shape separately, not integrating it comprehensively with appearance. As there has been development of other body shape reconstruction methods with implicit representation, integrating these reconstructions can further provide additional guidance to appearance in the input images.

2.4 3-D Shape Representation

The shape representation of a specific object is to record an object from the real world in a format that can be stored in a computer. Compared with single-frame images, 3-D shapes include the extra depth information and provide the part-related information that cannot be captured in 2-D images. In earlier works, researchers recorded the surface, such as the points as point clouds [143, 145], or faces as mesh [66], or the occupancy [125] of the object in the 3-D space to represent the 3-D object. However, such representations focus on the explicit shape of the object and lack perceptual understanding of the object and encode the coordinate with the object representation.

Due to the restrictions of recording shape with coordinates, recent works focus on decoupling the representation of shape and coordinates. Some researchers focus on object-specific feature representation as the implicit representation [136, 125, 21, 52, 152, 115, 126, 105, 217, 51, 55, 252] and include a decoder-like structure for extracting the explicit shape. To preserve more details of the objects and achieve higher quality for the rendered results, some researchers are projecting the representation to each point in space, as in neural radiance fields, or to part of the space, as in Gaussian Splatting.

As one of the 3-D shape representation, body shape reconstruction has seen significant improvement over recent years. Earlier efforts are primarily focused on explicit methods [119, 139] which include strong priors concerning body shape. However, recent research trends are shifting towards more advanced implicit representations, such as implicit functions [136, 38, 141, 152, 153, 193] and Neural Radiance Fields

(NeRF) [140, 206, 235, 77, 259]. Some approaches are now attempting to bridge between SMPL-based methods [119] and refine the final reconstruction results [215, 214, 259]. While most of these methods exhibit promising properties for fine-grained body shape reconstruction and rendering, as well as temporal consistency, their potential utility combining with appearance for downstream vision tasks remains largely unexplored. In addition, these methods are only able to capture per-scene rendering [129, 128, 263, 114, 134, 137, 140, 142, 259, 82, 102] or require an enormous amount of data as reference to render an image from a novel view [80, 224, 222, 192, 196, 117, 20]. Such restrictions make the appearance-based representation of body shapes not directly applicable to re-identification tasks.

Chapter 3

GaitSTR: Multimodal Gait Recognition

3.1 Introduction

In this section, we focus on one of the modality to identify a person: gait. Gait recognition [71, 170, 213, 225] is to identify the person present in a walking sequence. Different from other modalities, gait has the advantage of being able to be observed from a long distance and without the subject's cooperation. For gait recognition, researchers have developed silhouette-based methods, such as GaitSet [12], GaitPart [44], GaitGL [106], *etc.*, and skeleton-based methods like GaitGraph [180]. However, both input modalities exhibit certain deficiencies. Binarized silhouettes suffer from variations due to clothing and carried objects, as shown in Figure 3.1 (a), introducing external ambiguity, with segmented parts of a binarized silhouette being unavailable. Skeletons, on the other hand, include inconsistencies across frames in a sequence due to erroneous joint predictions, as depicted in Figure 3.1 (b), thereby reducing the accuracy of gait recognition.

We propose the fusion of silhouette sequences with skeletons, harnessing the advantages of both modalities by refining the skeletons using silhouette sequences. Given that jitters in the detected skeletons are confined to a few frames isolated from the entire sequence, they lack temporal consistency with their neighboring frames [230]. Simple temporal smoothing, however, can introduce further confusion for gait recognition as the generated skeletons create new poses inconsistent with the current sequence. On the other hand, silhouettes for neighboring frames exhibit better temporal consistency due to minor changes

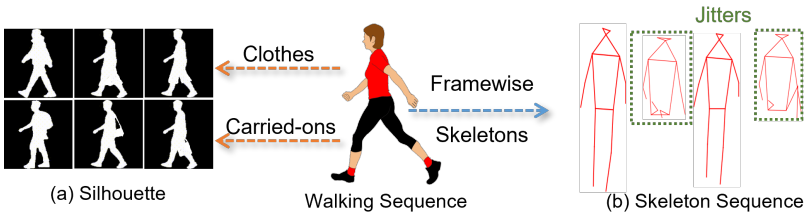


Figure 3.1: Visualization of the (a) silhouette and (b) skeleton sequence used for gait recognition. Silhouettes show different contours with different clothes and carried-on objects, while the skeletons suffer from jittery detection results in the video.

in adjacent image conditions. We enhance the quality of skeletons by employing silhouettes to rectify the jitters while retaining necessary identity information for more accurate gait recognition.

We introduce *GaitSTR*, a **S**equential **T**wo-stream **R**efinement method based to refine the skeletons and combine them with silhouettes for gait recognition in addition to feature aggregation methods, GaitMix and GaitRef [254]. When extracting the silhouette and skeletons from the same walking sequence, the temporal consistency between the two modalities is capable of providing guidance for each other: when silhouettes are not preserving useful boundaries of the images, skeletons can furnish the positions of the joints for pose estimation and recognition, and when the detection results of skeletons are unreliable, silhouettes can provide the pose information for the current frame in the sequence.

To refine the skeletons, we introduce two-level fusion: an internal fusion within skeletons and a cross-modal correction with the temporal guidance from the silhouettes describing the same walking sequences. As skeletons can be decoupled into two different representations [164, 257], joints, and bones, we incorporate self-correction between the frame-wise joints and bones for increased consistency. Introduction of the bones, in addition to joints, is to provide more connectivity as the GCN [219] primarily focuses on the position of the joints and does not explicitly explore the distances between the nodes other than binarized connectivity. To refine these two representations, we incorporate two different spatial-temporal graph convolution branches [219] for these two modalities, with the same number of layers and dimensions at each level. After each graph convolution operation, we utilize a self-correction residual block to forward

the information of the joints and bones, and add the information to the other branch residually [67] between the same level of layers.

In addition to the internal fusion of skeletons, we further introduce the cross-modal fusion between the silhouette and skeletons, combining the encoded silhouette features with the encoded frame-wise skeleton features to predict the relative changes for joints and bones in the skeletons. Since the gait pattern should be consistent for the same person, features from the silhouettes and skeletons describing the same walking sequence should also be consistent, facilitating the refinement of the skeletons with encoded silhouette features. Moreover, the sequence-level silhouette feature aids the frame-level skeletons for each frame in understanding its corresponding poses without losing identity information, as the temporal feature for the person is consistent and shared across all the frames in the same walking sequence.

With the predicted changes to the points, we reintegrate them into the original skeleton sequence and employ the skeleton encoder to extract the skeleton feature. We then concatenate this feature with the silhouette feature to predict the identity of the sequence using the refined skeletons. We compare *GaitSTR* with baseline multimodal gait recognition methods that use skeletons and silhouettes, including GaitMix [254], which concatenates the silhouette and skeleton features, and GaitRef [254], which uses silhouette features to refine joints. We show that skeleton refinement across the skeleton and silhouettes aids the final gait recognition, while adding internal correction within skeletons yields the best performance. We assess our method on four public datasets: CASIA-B [226], OU-MVLP [179], Gait3D [237], and GREW [262]. Our findings demonstrate that the refined skeletons, when combined with silhouettes, outperform other state-of-the-art gait recognition methods that utilize skeletons and silhouettes.

In summary, our contributions are: 1) we introduce *GaitSTR* which combines skeletons and silhouettes in an end-to-end training framework for gait recognition networks, 2) we incorporate the two different representations, joints and bones, for enhanced skeleton correction through self-fusion within skeletons,

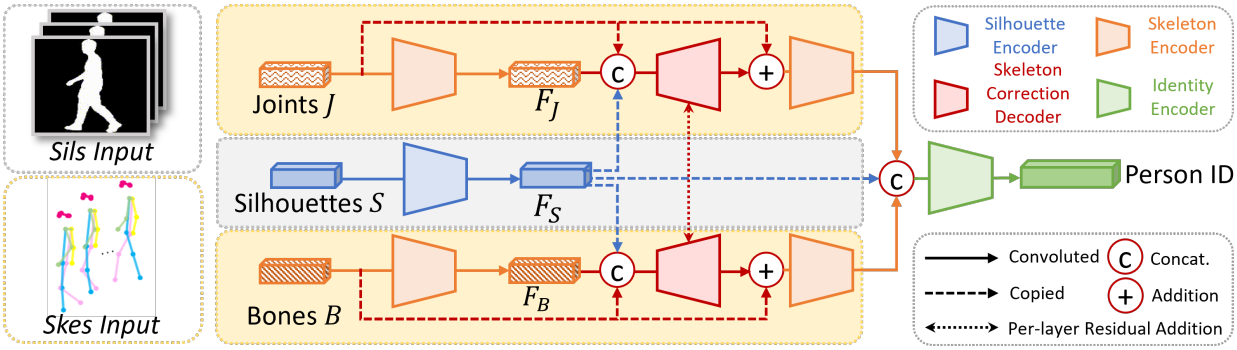


Figure 3.2: Our proposed architecture for GaitSTR. Trapezoids consists of trainable modules, and modules of the same color and fill-in patterns in the same model share the weight. Dashed lines represent the operation of feature copying. S, J, and B are the input silhouettes, joints, and bones, respectively. F_S represents silhouette features, while F_J and F_B represent joint and bone features for skeleton representations.

and 3) we utilize the temporal information in silhouette to assist in correcting jitters in skeletons without additional supervision.

This work is an extension of a conference version paper [254]. The novel contributions of this work are as follows.

1. In addition to using joints as the skeleton representations in GaitMix and GaitRef [254], we jointly utilize joints and bones as the representations. Unlike the single modal, the joints and bones represent different attributes of the skeletons, complementing each other.

2. Besides the cross-modal fusion between silhouette and skeletons for refinement, we also introduce fusion between joints and skeletons. We demonstrate that feature integration and refinement provide a more comprehensive understanding between each level of features, and yield more consistent feedback to the skeleton representation, which results in improved gait recognition accuracy.

3. We include additional experiments with *GaitSTR*, an extension of GaitRef, to analyze the contribution of skeletons and silhouettes in gait representation.

3.2 Method

To recognize the person's identity, we combine silhouettes and skeletons for recognition. For silhouettes, we use the binarized body boundary images as input, and for skeletons, we take both bones and joints into consideration. Motivated by the bones used in skeleton-based action recognition [257, 216], the introduction of bones focuses and emphasizes on the connections between the bones the joint-based graph convolution focuses majorly on the node instead of the connection between the nodes.

Given silhouettes S along with the joints J and bones B of the skeleton for the person p, the task of gait recognition is to match the identity with the people in a pool $P = \{p_n\}_{n=1,2,...}$, where n is the candidate identity. We encode S, J and B to their corresponding embeddings and concatenate them together to find the nearest sample in P in the embedding space. In this section, we first discuss the baseline combination the bones in addition to the joints and silhouettes in Section 3.2.1, and then we present our proposed method *GaitSTR* to refine the input skeleton for gait recognition in Section 3.2.2 along with objectives and details for training. We show our proposed architecture in Figure 3.2.

3.2.1 Multimodal Gait Recognition

We build the multimodal gait recognition model as our baseline to combine information from different input modalities, including skeleton, joint and silhouettes. We employ two encoders: a silhouette feature encoder designed for encoding the silhouette S, and a skeleton feature encoder tasked with projecting the dual representations of the input raw skeletons, namely joints J and bones B, into their corresponding embedding spaces. Features generated from these two encoders are concatenated together and used for gait recognition.

Silhouette Feature Encoder. To extract the identity features from input sequential silhouette sequences, we use a silhouette feature encoder to convert the input silhouette sequence S to the corresponding output identity feature F_S. We have three steps for the silhouette feature encoder: convolution

feature extraction, temporal pooling, and horizontal pooling. With the binary silhouette input sequence $S = \{s_i\}_{i=1,..,N}$, where i is the temporal stamp and N is the overall frame number, we apply a convolution network to extract the framewise feature fi at frame i. f_i is an M-by-N-by-C matrix, where M and N are the height and width of the convoluted output features, and C is the channel number from the output of the last convolution layer.

With the framewise feature f_i, we use a max pooling layer for the temporal fusion and combine the feature into a single M-by-N-by-C output as temporal pooling. Since f_i still includes the spatial features for each segment, we follow [48] and apply horizontal pyramid pooling with scale S as 5. The output of the feature is a 2^{S-1}-by-C feature vector after horizontal pooling. The architecture of each component can be found in the implementation details in Section 3.3.1.

Skeleton Feature Encoder. In addition to the silhouette encoder, we deploy a skeleton feature encoder in parallel. This encoder processes the input skeleton sequences, consisting of joints $J = \{j_i\}_{i=1,...,N}$ that record the position of each keypoint of the body, and bones $B = \{b_i\}_{i=1,...,N}$ that are represented as vectors denoting the numerical directional relationship between two connecting joints, into their associated embeddings denoted as F_J and F_B. To represent both the joint and bone branches, we utilize two identical graph convolution networks; however, these networks do not share weights.

With the N-by-K-by-2 matrix to depict the 2-D skeletons of each frame, where K (either K_J for joints or K_B for bones) represents the number of points or connections of the skeletons, we implement a multi-layer spatial-temporal graph convolution network [219] for graphical feature extraction for each of them. By transforming the input from dimensions N-by-K-by-2 to N-by-K-by-C as the pre-frame per-node feature matrix, we then conduct average pooling over both the temporal and node dimensions and produce two final C-length vectors. These vectors, F_J and F_B, represent the features of the sequential skeleton pertaining to the input joints and bones, respectively. The two vectors are aggregated together

via concatenation as the 2-by-C feature output of the skeleton feature encoder, and then concatenated with silhouette features to represent the input sequence for recognition.

3.2.2 GaitSTR: Sequential Two-stream Refinement

In addition to fusing features from skeletons and silhouettes for gait recognition, *GaitSTR*, an extension of *GaitRef* [254] introduces encoded features from silhouettes to improve the temporal consistency of skeletons, thereby enhancing the quality of skeletal data. The consistency across the two representations addresses framewise jitters in skeleton generation, ensuring a smoother and more accurate skeletal representation. Conversely, refined skeletons contribute to the silhouette analysis by minimizing the impact of appearance variants on gait recognition. Besides the two primary encoders employed for multimodal gait recognition, in *GaitRef* [254], we incorporate an additional skeleton correction network to correct the joint jitters in the skeleton sequences using the temporal consistency between skeletons and silhouette representations. In addition, *GaitSTR* further mines the consistency between the joint and bone representations using a cross-modal adapter, which bridges the gap between joint and bone representations to bolster the robustness of gait recognition.

Skeleton Correction Network. With information from the joint and bone feature, we use three different features as the network's input to correct the skeleton and compute the corresponding adjustment for each point: 2^{S-1}-by-C silhouette features F_S, N-by-K-by-C skeleton feature before pooling, and the original N-by-K-by-2 joint or bone matrix J or B. F_S provides the sequential information to correct the joint features F_J and F_B. F_J and F_B provide the framewise and feature for each node to correct the corresponding position of the joint in the frame. J and B provide the input order of the points to ensure the input and output order of the points are the same.

We show the architecture of the skeleton correction network in Figure 3.3. With these three inputs, we first flatten the silhouette feature into a $2^{S-1} \times C$ vector. We then repeat it N-by-K times and concatenate

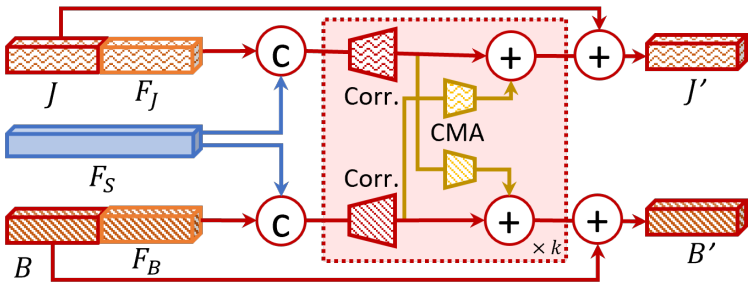


Figure 3.3: Architecture of the skeleton correction network. F_J and F_B represent the joint and bone frame-wise features encoded from J (joints) and B (bones), respectively. The symbol 'C' denotes concatenation, and the plus sign denotes addition. 'Corr' refers to the skeleton correction network, while 'CMA' stands for the layer-level cross-modal adapter, and k denotes the number of layers over which cross-modal skeleton correction operations are repeated between bones and joints.

it with the other two features to form a N-by-K-by-$(2^{S-1} \times C + C + 2)$ feature matrix. To decode the new position J' for each node in the sequence, we decode the ΔJ for all the points with a reversed spatial-temporal graph convolution network to decode the N-by-K-by-2 adjustment for each node in J, and we have J' for refine the individual points in J following

$$J' = J + \Delta J = J + SkeletonDecoder(J, F_S, F_J). \tag{3.1}$$

The use of addition instead of directly predicting the corresponding location of the refined joints can give a relatively easier task for refinement and can preserve most of the original locations [253], since the original position of the joint has most of the sequential information correct and complete. By adding ΔJ on J, we get the final refined nodes as output and process it for further encoding.

Likewise, in the bone stream, we employ the same correction network to produce the adjustments, denoted as ΔB, on the original bone matrix B. This refinement follows a similar procedure as in the joint stream, formulated as:

$$B' = B + \Delta B = B + SkeletonDecoder(B, F_S, F_B). \tag{3.2}$$

With this refinement, we obtain the refined bone matrix B' with the assistance of encoded silhouette features, which is then utilized for further encoding in the network.

Cross-Modal Adapter. As the bones and joints represent the same skeleton and have connections between them, refining the skeleton and joints are also relevant to each other. We introduce the cross-modal adapter, $\text{CMA}_{i \to j}$, between these two modalities, as illustrated in Figure 3.3, where i and j represent the source and target modalities. As the correction network includes multiple layers as input and their architectures are similar, for each decoded skeleton representation F_B^x and F_J^x at layer x, we employ a two-layer MLP to project the features for the other modality following

$$\begin{aligned} F_B^x &= F_B^x + CMA_{J \to B}(F_J^x) \\ F_J^x &= F_J^x + CMA_{B \to J}(F_B^x). \end{aligned} \tag{3.3}$$

The refined feature F_B^x will be used for the input of the next graph convolution layer $(x+1)$ to enhance inter-modal communication for more robust and accurate gait recognition.

After we get the refined skeleton J' and B', we apply the same skeleton feature encoder in Section 3.2.1 and apply it on the refined skeleton sequence for predicting the 1-by-C skeleton feature $F_{J'}$ and $F_{B'}$. The two skeleton feature encoders for each modality share the parameters to ensure the two embedding spaces are the same between F_J and $F_{J'}$ as well as between F_B and $F_{B'}$. Using the same skeleton feature encoder can also extend the data available for the encoder training to train a stabler graph convolution model the skeleton feature extraction.

With the predicted $F_{J'}$ and $F_{B'}$, we concatenate them with 2^{S-1}-by-C silhouette feature F_S for representing the human body shape for *GaitSTR*. In addition, we also include the joint feature before refinement as the final representation. The use of a combination of both F_J and $F_{J'}$ ensures that, during training, the network can maximize its ability to distinguish the identities from the skeletons. In addition, using both

of the features encoded from the two skeleton sequences gives the most representation for the task of gait recognition.

3.2.3 Objectives and Inference

We have include two losses for training *GaitSTR*: a triplet loss $L_{triplet}$ for distinguishing the same identities in the same batch and a classification loss L_{cls} for the identities in training set with an MLP layer for projecting the identity feature to the number of candidates. For the combination of the two losses, we follow

$$L = \lambda_1 L_{triplet} + \lambda_2 L_{cls} \tag{3.4}$$

and empirically set λ_1 as 1. For λ_2 we follow [106, 237] to set it as different values for different datasets. We include further discussion and the choice of parameters in the implementation details section in Section 3.3.1.

3.3 Experiments and Results

In this section, we first discuss the details of our experiment settings, including datasets, implementation details, and baseline methods in Section 3.3.1, followed by our numerical and visualization results in Section 3.3.2.

3.3.1 Experimental Details

Datasets. In our experiment, we assess our method on four public gait recognition datasets, CASIA-B [226], OUMVLP [71, 3], Gait3D [237] and GREW [262].

CASIA-B [226] has 124 subjects with 10 different walking variants for gait recognition. Among the 10 variants, 6 variants are for normal walking (NM), 2 variants are for the person carrying different bags (BG),

and the remaining 2 variants are for different clothes (CL). Each subject has 110 videos captured with 10 variants from 11 different camera viewpoints distributed between 0and 180 We follow [12, 44, 76, 106] and use the videos of the first 74 identities for training and the remaining 50 for inference. During inference, we use the first four variances in normal walking conditions (NM) to build the gallery set as the library to query test sequences. The sequences of the remaining 2 NM variants, along with BG and CL sequences, are used as probe examples for finding the identity in the gallery.

OUMVLP [179, 3] is a large-scale dataset with 10,307 different identities. Each subject in this dataset has 2 different variants for normal walking (NM) conditions from 14 camera viewpoints, making 28 gait sequences. The angles of camera viewpoints are evenly distributed in two bins, 0to 90 and 180to 270 Every two neighbor viewpoints have a 15-degree gap. We follow [12, 44, 76, 106] to use the identities with odd indexes between the 1-*st* and 10,305-*th* examples and build a training set with 5,153 identities. For the remaining 5,154 identities, we use the first sequence as the gallery set and the second as probes during inference.

Gait3D [237] is a medium dataset compared with CASIA-B and OUMVLP for gait recognition in the wild. It includes 4,000 identities among 25,309 video sequences captured via 39 cameras. Since sequences are captured in the wild, camera positions, carried-on objects, and clothes vary from sequence to sequence. Similar to GREW [262], Gait3D also provides both skeletons and silhouette sequences for each frame in the dataset. We follow [237] to use 3,000 identities for training and the remaining 1,000 during inference. For these 1,000 test cases, we build a probe set with 1,000 sequences for querying, as the probe set, and use the rest 5,369 sequences as the gallery set.

GREW [262] is a large in-the-wild gait recognition dataset with 128,671 sequences capturing 26,345 identities from 882 cameras. Each frame in the video has both silhouettes and poses provided. We follow [262] for using 20,000 identities for training and 6,000 identities as our test set. Each subject in the

test set has 4 sequences, where we use two for the gallery and the other two as probe videos following the official split [262].

Implementation Details. For the implementation details section, we will discuss the details for the data preparation, model, and hyperparameter selection in experiments.

Data preparation. For all four datasets, we follow OpenGait* to prepare the silhouettes for each dataset, setting the size of each frame to 64×44. Unlike silhouettes, skeletons provided for different datasets vary. Thus, we process the joints for each dataset independently. For the CASIA-B [226] dataset, we follow GaitGraph [180] and use a pretrained HR-Net [174] to generate the skeleton in MS COCO [109] format with 17 joints. The number of frames used for the skeletons of CASIA-B is set to 60, using the 60 frames at the center of the entire sequence as joint input.

For OUMVLP [179] dataset, we follow [3] for applying the skeletons along with the silhouette sequences, and we have skeleton sequences with 18 nodes per frame as OpenPose [7] format. Considering that the sequence length in OUMVLP is shorter than CASIA-B, we set the fixed frame number to 25 for each sequence. For videos shorter than 25, we repeat the frames until we have 25 frames.

For Gait3D [237] and GREW [262], since skeletons are collected in the wild, we normalize each skeleton by setting their height to 2 and move their center to the origin point $(0, 0)$. This can ensure that the position of the skeletons is aligned across different videos and will not change significantly.

In addition to joints, we generate bones based on predefined neighbor link relationships between joints, represented as directional vectors calculated from the differences in coordinates between linked joints. Compared to the joints in the skeletons, the connected bones are defined by neighbor link relationships, emphasizing the numerical connectivity that is not explicitly captured in ST-GCN [219].

Network details. In our network, we have two different encoders. For our silhouette feature encoder, we follow GaitGL [106] to build the encoder for CASIA-B. For Gait3D, OUMVLP, and GREW, we follow

*https://github.com/ShiqiYu/OpenGait

GaitBase [43] to encode silhouette features. Note that for GaitMix and GaitRef, we follow [254] to use OpenGait [43] for Gait3D and GaitGL [106] for GREW respectively. For the skeleton feature encoder, we follow ST-GCN [219] for encoding the skeletons into the same embedding dimension N_{out} as the silhouette feature encoder. The dimension of the hidden layers of ST-GCN is set to [64, 64, 128, 128, n_{out}]. The two skeleton decoders of the *GaitSTR* both use the reversed shape of the ST-GCN, with [128, 64, 64, 3] as the hidden dimensions and the number of CMA, K is set to 3. For the encoder and decoder network, we have compared ST-GCN along with other choices, such as MS-G3D [118] for ablation study.

Model training. In our model, we follow [106, 43] for choosing the hyperparameters. For CASIA-B, we use an Adam optimizer [91] with $1e-4$ as the learning rate for 80,000 iterations. We decay the learning rate once at 70,000 iterations for CASIA-B as $\frac{1}{10}$ of its original value. For Gait3D, OUMVLP and GREW, we use the SGD optimizer for 60,000, 120,000 and 180,000 iterations, respectively and set the initial learning rate as $1e-3$. The learning rate is decayed to $\frac{1}{10}$ three times for these three datasets, at iteration 20,000, 40,000 and 50,000 for Gait3D, 60,000, 80,000 and 100,000 for OUMVLP, and 80,000, 120,000 and 150,000 for GREW. For all four datasets we use, we follow [106, 43] for using 1 for both λ_1 and λ_2 in our experiment.

Metrics and evaluations. During inference, for each example in the probe set, we use L_2 similarity to find the nearest example in the gallery set. For CASIA-B and OUMVLP, we evaluate the top-1 accuracy for the prediction. For GREW, we evaluate top-1, 5, 10 and 20 accuracies. For Gait3D, we assess top-1 and top-5 accuracies along with mAP and mINP following [223] for assessing since all the correct matches should have low-rank values when pairing the probe example with correct identities in the gallery.

For baseline methods, we compare with state-of-the-art approaches, including CNN-LB [213], GaitNet [170], GaitSet [12], GaitPart [44], GLN [76], GaitGL [106], ModelGait [99], and CSTL [78]. Additionally, we compare with PoseGait [104] and GaitGraph [180], which utilize skeleton sequences as their input. For baseline comparison, we include GaitMix and GaitRef [254], which use simple concatenation as described in Section 3.2.1 and only utilize the skeleton correction network as outlined in Section 3.2.2, respectively.

GaitSTR, along with these methods, employs 2-D convolution for binarized silhouette feature extraction. We also include comparisons with other methods which use different modalities, such as GaitEdge [103], which generates silhouettes from RGB images, and MvModelGait [98], which requires RGB images and camera positions, as well as methods using more complex silhouette encoder like TriGait [176], which uses 3-D convolution in its backbone.

3.3.2 Results and Analysis

In this subsection, we first present the numerical results for CASIA-B [226], OUMVLP [71, 3], Gait3D [237], and GREW [262] compared to other state-of-the-art methods. We then delineate and analyze the enhancements conferred by the multi-modal gait model as opposed to the refinement of the skeletons from the silhouettes. In addition, we compare the use of skeletons with 3-D body shapes on the top of silhouettes for gait recognition. Finally, we present an ablation study for our model and visualizations that illustrate the corrected skeletons informed by silhouette guidance.

Numerical Results. We present our numerical performance on the four datasets used in our experiments in Table 3.1, 3.2, 3.3, and 3.4, respectively. We follow the official splits of these four datasets for gallery and probe constructions. For CASIA-B and OUMVLP, identical-view cases are excluded.

For all four datasets evaluated, we outperform the existing state-of-the-art methods with *GaitSTR*. In Table 3.1, on CASIA-B, we achieve the best performance on all splits. Specifically, on NM, BG, and CL, we reduce the error rates from 2.1%, 5.6%, and 15.8% to 1.6%, 3.8%, and 10.4%, respectively, which correspond to a relative reduction of 23.8%, 32.1%, and 34.2% in error rates compared with the best model using 2-D convolution for silhouette feature extraction, while we also show similar performance compared to methods [176] using 3-D convolution. The margin of improvement is even greater for NM and CL settings when compared to our baseline silhouette encoder, GaitGL, where we demonstrate a 33.3% and 37.0% relative reduction for the average rank-1 predictions across all camera views. Furthermore,

Table 3.1: Gait recognition results on CASIA-B dataset, excluding identical-view cases. TriGait includes a 3-D convolution feature extractor which requires much heavier computation than the 2-D encoders used by other methods in the table. We mark the best results among all the methods in bold and the best results in our baseline methods with underline.

Probe	Method	Camera Positions											Mean
		0	18	36	54	72	90	108	126	144	162	180	
NM #5-6	CNN-LB [213]	83.3	92.3	96.7	94.6	91.7	89.7	92.2	94.0	96.3	92.3	79.0	91.1
	GaitNet [170]	91.2	92.0	90.5	95.6	86.9	92.6	93.5	96.0	90.9	88.8	89.0	91.6
	GaitGraph [180]	85.3	88.5	91.0	92.5	87.2	86.5	88.4	89.2	87.9	85.9	81.9	87.7
	GaitSet [12]	91.1	98.0	99.6	97.8	95.4	93.8	95.7	97.5	98.1	97.0	88.2	95.6
	GaitPart [44]	94.0	98.7	99.3	98.8	94.8	92.6	96.4	98.3	99.0	97.4	91.2	96.4
	GLN [76]	93.8	98.5	99.2	98.0	95.2	92.9	95.4	98.5	99.0	99.2	91.9	96.5
	GaitGL [106]	95.3	97.9	99.0	97.8	96.1	95.3	97.2	98.9	99.4	98.8	94.5	97.3
	CSTL [78]	97.2	99.0	99.2	98.1	96.2	95.5	97.7	98.7	99.2	98.9	96.5	97.8
	ModelGait [99]	96.9	97.1	98.5	98.4	97.7	98.2	97.6	97.6	98.0	98.4	98.6	<u>97.9</u>
	GaitMix [254]	96.6	98.6	99.2	98.0	97.1	96.2	97.5	98.9	99.3	99.0	94.7	97.7
	GaitRef [254]	97.2	98.7	99.1	98.0	97.3	97.0	98.0	99.4	99.4	98.9	96.4	98.1
	GaitSTR	97.2	98.4	99.2	98.3	97.6	97.8	97.9	99.3	99.3	99.3	97.6	**98.4**
	MvModelGait [98]	97.5	97.6	98.6	98.8	97.7	98.9	98.9	97.3	97.6	97.8	97.9	98.1
	GaitEdge* [103]	97.2	99.1	99.2	98.3	97.3	95.5	97.1	99.4	99.3	98.5	96.4	97.9
	TriGait [176]	97.0	98.6	98.3	98.3	98.4	97.0	98.6	99.0	98.9	98.4	97.4	98.2
BG #1-2	CNN-LB [213]	64.2	80.6	82.7	76.9	64.8	63.1	68.0	76.9	82.2	75.4	61.3	72.4
	GaitNet [170]	83.0	87.8	88.3	93.3	82.6	74.8	89.5	91.0	86.1	81.2	85.6	85.7
	GaitGraph [180]	75.8	76.7	75.9	76.1	71.4	73.9	78.0	74.7	75.4	75.4	69.2	74.8
	GaitSet [12]	87.0	93.8	94.6	92.9	88.2	83.0	86.6	92.6	95.7	92.9	83.4	90.1
	GaitPart [44]	89.5	94.5	95.3	93.5	88.5	83.9	89.0	93.6	96.0	94.1	85.3	91.2
	GLN [76]	92.2	95.6	96.7	94.3	91.8	87.8	91.4	95.1	96.3	95.7	87.2	93.1
	GaitGL [106]	93.0	95.7	97.0	95.9	93.3	90.0	91.9	96.8	97.5	96.9	90.7	<u>94.4</u>
	CSTL [78]	91.7	96.5	97.0	95.4	90.9	88.0	91.5	95.8	97.0	95.5	90.3	93.6
	ModelGait [99]	94.8	92.9	93.8	94.5	93.1	92.6	94.0	94.5	89.7	93.6	90.4	93.1
	GaitMix [254]	94.4	96.7	96.8	96.1	94.3	90.4	93.5	97.4	98.0	97.2	92.2	95.2
	GaitRef [254]	94.4	96.4	97.3	96.8	96.2	92.2	94.4	97.2	98.7	97.9	93.3	95.9
	GaitSTR	95.3	97.1	97.8	96.8	96.1	93.2	94.3	96.8	98.3	98.3	94.0	**96.2**
	MvModelGait [98]	93.9	92.5	92.9	94.1	93.4	93.4	95.0	94.7	92.9	93.1	92.1	93.4
	GaitEdge* [103]	95.3	97.4	98.4	97.6	94.3	90.6	93.1	97.8	99.1	98.0	95.0	96.1
	TriGait [176]	91.8	94.3	95.2	96.6	96.5	93.7	95.9	97.6	97.4	96.9	93.8	95.4
CL #1-2	CNN-LB [213]	37.7	57.2	66.6	61.1	55.2	54.6	55.2	59.1	58.9	48.8	39.4	54.0
	GaitNet [170]	42.1	58.2	65.1	70.7	68.0	70.6	65.3	69.4	51.5	50.1	36.6	58.9
	GaitGraph [180]	69.6	66.1	68.8	67.2	64.5	62.0	69.5	65.6	65.7	66.1	64.3	66.3
	GaitSet [12]	71.0	82.6	84.0	80.0	71.7	69.1	72.1	76.7	78.5	77.2	63.4	75.1
	GaitPart [44]	72.5	82.8	86.0	82.2	79.5	71.0	77.7	80.8	82.9	81.4	67.7	78.6
	GLN [76]	78.5	90.4	90.3	85.1	80.2	75.8	78.1	81.8	80.9	83.2	72.6	81.5
	GaitGL [106]	71.7	90.5	92.4	89.4	84.9	78.1	83.1	87.5	89.1	83.9	67.4	83.5
	CSTL [78]	78.1	89.4	91.6	86.6	82.1	79.9	81.8	86.3	88.7	86.6	75.3	<u>84.2</u>
	ModelGait [99]	78.2	81.0	82.1	82.8	80.3	76.9	75.5	77.4	72.3	73.5	74.2	77.6
	GaitMix [254]	79.2	89.5	94.2	90.0	84.9	80.3	85.2	89.2	90.3	86.9	73.7	85.8
	GaitRef [254]	81.4	93.3	94.3	91.6	87.8	83.9	88.5	91.7	91.6	89.1	75.0	88.0
	GaitSTR	83.8	94.0	94.9	94.3	90.7	85.5	89.2	91.8	92.8	90.7	78.2	**89.6**
	MvModelGait [98]	77.0	80.0	83.5	86.1	84.5	84.9	80.6	80.4	77.4	76.6	76.9	80.7
	GaitEdge* [103]	84.3	92.8	94.3	92.2	84.6	83.0	83.0	87.5	87.4	85.9	75.0	86.4
	TriGait [176]	91.7	93.2	96.9	97.0	95.2	94.0	94.6	95.3	94.1	94.1	90.8	94.3

Table 3.2: Gait recognition results for accuracy across all the test views on OUMVLP dataset, excluding identical-view cases.

Method	Camera Positions														Mean
	0	15	30	45	60	75	90	180	195	210	225	240	255	270	
GEINet [166]	23.2	38.1	48.0	51.8	47.5	48.1	43.8	27.3	37.9	46.8	49.9	45.9	45.7	41.0	42.5
GaitSet [12]	79.2	87.7	89.9	90.1	87.9	88.6	87.7	81.7	86.4	89.0	89.2	87.2	87.7	86.2	87.0
GaitPart [44]	82.8	89.2	90.9	91.0	89.7	89.9	89.3	85.1	87.7	90.0	90.1	89.0	89.0	88.1	88.7
GLN [76]	83.8	90.0	91.0	91.2	90.3	90.0	89.4	85.3	89.1	90.5	90.6	89.6	89.3	88.5	89.2
GaitGL [106]	84.2	89.8	91.3	91.7	90.8	91.0	90.4	88.1	88.2	90.5	90.5	89.5	89.7	88.8	89.6
CSTL [78]	87.1	91.0	91.5	91.8	90.6	90.8	90.6	89.4	90.2	90.5	90.7	89.8	90.0	89.4	90.2
GaitBase [43]	87.2	91.2	91.8	92.0	91.4	91.2	90.8	88.9	90.4	91.1	91.3	90.7	90.5	90.0	90.6
GaitMix [254]	85.4	90.3	91.2	91.5	91.2	90.9	90.5	88.9	88.7	90.3	90.5	89.8	89.6	88.9	89.9
GaitRef [254]	85.7	90.5	91.6	91.9	91.3	91.3	90.9	89.3	89.0	90.8	90.8	90.1	90.1	89.5	90.2
GaitSTR	87.6	91.5	91.8	92.1	91.5	91.3	91.0	89.2	90.7	91.1	91.3	90.8	90.6	90.2	**90.8**

Table 3.3: Gait recognition results reported on the Gait3D dataset with 64×44 as input sizes. For all four metrics, higher values of the same metric indicate better performance. B represents the bone input.

Methods	Rank@1	Rank@5	mAP	mINP
GaitSet [12]	36.70	58.30	30.01	17.30
GaitPart [44]	28.20	47.60	21.58	12.36
GLN [76]	31.40	52.90	24.74	13.58
GaitGL [106]	29.70	48.50	22.29	13.26
OpenGait [237]	42.90	63.90	35.19	20.83
CSTL [78]	11.70	19.20	5.59	2.59
GaitBase [43]	62.00	78.80	53.17	35.33
SMPLGait [237]	46.30	64.50	37.16	22.23
GaitMix [254]	46.20	66.20	37.08	22.85
GaitRef [254]	49.00	69.30	40.69	25.26
GaitSTR	**65.10**	**81.30**	**55.59**	**36.84**

when compared with GaitEdge [103] and MvModelGait [98], which utilize RGB images and viewpoint angles not typically present in public datasets, *GaitSTR* still exhibits superior performance, indicating the effectiveness of using the skeletons along with silhouettes is able to outperform the methods directly using the RGB images as input, which actually include all the information stored in the silhoeuttes and skeletons.

For the other three datasets, on OUMVLP in Table 3.2, we show small improvement compared with GaitBase [43] for the top-1 accuracy, while we outperform it along with other methods for all the metrics on Gait3D [237] and GREW [262] in Table 3.3 and 3.4, which we show 3.1% and 3.9% improvements on rank-1 accuracies respectively compared with other state-of-the-art methods using 2-D convolution as

Table 3.4: Rank-1, 5, 10 and 20 accuracies on GREW dataset.

Methods	Rank-1	Rank-5	Rank-10	Rank-20
PoseGait [104]	0.2	1.1	2.2	4.3
GaitGraph [180]	1.3	3.5	5.1	7.5
GEINet [166]	6.8	13.4	17.0	21.0
TS-CNN [213]	13.6	24.6	30.2	37.0
GaitSet [12]	46.3	63.6	70.3	76.8
GaitPart [44]	44.0	60.7	67.4	73.5
CSTL [78]	50.6	65.9	71.9	76.9
GaitGL [106]	51.4	67.5	72.8	77.3
GaitBase [43]	60.1	75.5	80.4	84.1
GaitMix [254]	52.4	67.4	72.9	77.2
GaitRef [254]	53.0	67.9	73.0	77.5
GaitSTR	**64.0**	**78.5**	**83.2**	**86.3**

silhouette feature encoders. We also show improvement on other metrics, such as mAP and mINP [237] used by Gait3D [237].

Improvement of Skeleton Refinement. With the inclusion of both bones and joints to represent skeletons, we further analyze the improvement from introducing a new modality and the manner in which these modalities are utilized in *GaitSTR*. We present the results in Table 3.5, showing rank-1 accuracy on CASIA-B [226]. We begin with our baseline method, GaitGL [106], which operates on a single modality, and then proceed to analyze the introduction of joints and bones, as well as various ways of integrating them with silhouettes for different combinations.

When comparing the use of silhouettes as the only input for gait recognition, the introduction of joints as skeletons displays an improvement for both simple feature correction, as seen with GaitMix [254], and the use of silhouettes to refine joints, as with GaitRef [254] across all three metrics. GaitRef, which uses silhouettes to refine the joints, provides better recognition accuracy compared to simply aggregating the two features.

Furthermore, the introduction of bones in both GaitMix and GaitRef leads to additional improvements over the use of joints alone with silhouettes. For GaitRef, we treat the newly introduced bones similarly

to joints and utilize silhouette features to refine them. We then concatenate the three encoded features (silhouettes, joints, and bones) for recognition. Including a single-sided bone-to-joint cross-modal adapter results in a slight decrease in performance, whereas incorporating adapters on both sides, as in *GaitSTR*, shows better performance. The refinement from one side creates inconsistencies across the two skeletal representations, while the two-way sequential refinement provides consistent enhancement across these modalities.

Skeleton and Body Shape for Gait Recognition. Gait3D [237] provides 3-D body shapes alongside silhouette sequences, which are utilized by SMPLGait [237]. In Table 3.3, we provide a comparison of using 3-D body shapes as in SMPLGait [237] and using skeletons by removing the skeleton correction network and cross-modal adapters, which directly aggregate skeleton features with joint features as in GaitMix [254] and GaitRef [254]. For a fair comparison, we use GaitMix and GaitRef [254] with the OpenGait [43] baseline without augmentation, as SMPLGait [262] is not implemented on the latest OpenGait configuration with data augmentation.

Compared to using silhouettes as the only input modality, the inclusion of skeletons and body shapes both enhance recognition accuracy. In SMPLGait, skeleton information is partially integrated into the generated 3-D body shapes for gait recognition, making SMPLGait yields similar performance as GaitMix [254], which includes joints and bones, across all four metrics.

When compared to SMPLGait, which uses a 3-D body shape as the second modality, GaitRef [254] with bone inputs for the refined skeletons achieves better recognition performance. Considering that the generation of SMPL body shapes also requires skeletons [178], inaccurate pose estimation in 3-D body shape generation can hinder the model's ability to correctly interpret noisy body shapes with erroneous poses in SMPLGait [237]. GaitRef, however, does not suffer from this issue with refined skeletons.

Ablation Studies. For ablation studies, we present results on: 1) different methods of combining skeleton and silhouette features, 2) various skeleton encoder and decoder networks in comparison with

Table 3.5: Rank-1 accuracy of the variations skeletons in addition to silhouettes for gait recognition on CASIA-B. 'Sils.' represents silhouettes.

Input Modality	Methods	NM	CL	BG
Sil. only	GaitGL [106]	97.3	94.4	83.5
Sil. + Joint	GaitMix [254]	97.7	95.2	85.8
	GaitRef [254]	98.1	95.9	88.0
Sil. + Joint + Bone	GaitMix (w/ Bone)	98.0	95.6	87.5
	GaitRef (w/ Bone)	98.2	96.0	88.9
	+ $\text{CMA}_{B\rightarrow J}$	98.1	95.7	88.7
	GaitSTR	**98.4**	**96.2**	**89.6**

Table 3.6: Ablation results for different silhouette and skeleton feature combination on CASIA-B dataset for three splits. 'Padding' indicates the skeleton feature is padded on each of the feature of different scales, while 'concat.' means we concatenate the feature along with the scale dimension and use it only once.

Method	Combination	NM	CL	BG
GaitGL [106]	N/A	97.3	94.4	83.5
Sil. + Joints	Padding	97.5	94.6	85.8
Sil. + Joints	Concat.	**98.1**	**95.9**	**88.0**

other skeleton refinement methods, and 3) the inputs of the skeleton correction network. All experiments were conducted on the CASIA-B dataset [226] for each of the three different settings, and we present the Top-1 accuracy for the final gait recognition results. The results are shown in Tables 3.6, 3.7, and 3.8, respectively. Since the joint and bones branches are identical and exhibit similar performance, our ablation experiments focus on the joints branch as skeleton representation.

(i) Feature Combination. In addition to concatenating the features, we also repeat and pad the skeleton feature along with each segment of the silhouette features to provide the guidance for different level of silhouette embeddings, which we label as 'padding'. We show the results in Table 3.6. For comparison, we also add the performance of GaitGL [106] in the table, which only uses the silhouette feature for gait recognition and is our backbone baseline on CASIA-B. We observe that padding the skeleton feature

Table 3.7: Ablations for different encoder and decoder combinations for silhouette with joints and different skeleton smoothing methods on CASIA-B datasets. Results are reported in Top-1 accuarcy.

Encoder	Decoder	NM	CL	BG
ST-GCN	N/A	97.7	95.2	85.8
MS-G3D	N/A	98.0	95.5	86.4
ST-GCN	ST-GCN	**98.1**	**95.9**	88.0
ST-GCN	MS-G3D	**98.1**	95.7	**88.5**
MS-G3D	ST-GCN	**98.1**	**95.9**	88.3
Average Smoothing		97.6	95.0	85.6
Gaussian Smoothing		97.7	95.2	85.9
SmoothNet [230]		97.4	94.4	83.8

alongside each size of the silhouette feature results in worse performance compared to concatenating the refined feature just once as the final recognition feature. Padding the skeleton feature multiple times may cause the skeleton input to dominate the feature space, whereas concatenating it once allows the silhouette features to contribute robust information about the pose and remain less sensitive to skeletons.

(ii) Encoder-decoder Variations. For the choice of the skeleton encoder and skeleton correction decoder, we select between two state-of-the-art skeleton action recognition models: ST-GCN [219] and MS-G3D [118]. The results are presented in Table 3.7. Following previous observations, where using joints as the skeleton representation follows the same trend as using both joints and bones, we opt for using only joints as the skeleton representation for this ablation. Both MS-G3D and ST-GCN show improvement in performance. However, in our experiments, MS-G3D requires significantly more GPU memory and at least double the training time for each module introduced. Considering their comparable performance and time efficiency, we choose ST-GCN for both the encoder and decoder in our final pipeline.

(iii) Skeleton Refinement. For skeleton refinement, we compare the refining of the skeleton sequence using silhouettes with neighbor smoothing (average and Gaussian window for the neighboring three frames) and SmoothNet [230] (pretrained on H36m [79]) on the CASIA-B dataset based on Top-1 accuracy. The results, displayed in Table 3.7, demonstrate that refining skeleton with silhouettes outperforms

Table 3.8: Ablation results of different input for the skeleton correction network on CASIA-B. SCN is skeleton correction network.

Corr. Input	NM	BG	CL
w/o F_J	97.7	95.4	87.0
w/o F_S	97.6	95.3	85.6
w/o J	97.3	95.5	86.0
Full SCN	**98.1**	**95.9**	**88.0**

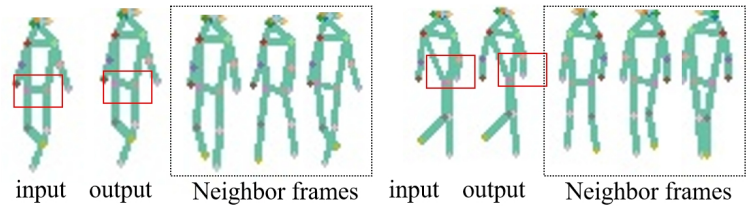


Figure 3.4: Visualization of successful and failure refined skeletons with *GaitSTR*. For each example, from left to right, we have original skeletons, refined skeletons and its neighbor frames.

the other methods. Among the three variations, 3-frame Gaussian smoothing shows a slight improvement but still falls short compared to using silhouettes.

Different from naive temporal smoothing, which can result in poses inconsistent in the sequence, integrating silhouette features introduces walking patterns not present in the skeletons, aiding their self-refinement for gait recognition. Compared to skeletons refined from the skeleton sequence alone, external knowledge from encoded silhouette embeddings reduces ambiguity, providing ID-specific information during training when the walking pattern cannot be correctly extracted from the skeleton alone.

(iv) Input of the Skeleton Correction Network. Considering three distinct inputs in our skeleton correction network, F_J, F_S, and J, we investigate each component's contributions and present the results in Table 3.8 using three splits of the CASIA-B datasets. We note that when either of the three input is excluded, there is a significant drop in performance. The skeleton correction network capitalizes on temporal consistency in the skeleton sequences for correction, while the additional silhouette information provides external support for an enhanced understanding.

Skeletons Visualization. We present two examples from *GaitSTR* compared to the original skeletons in Figure 3.4, accompanied by the three nearest silhouettes from a similar timestamp. With two modalities of representations for gait, *GaitSTR* can make more precise modifications to the skeletons' nodes. However, it still fails on some obvious errors, as seen in the second example in Figure 3.4.

3.4 Conclusion

We introduce *GaitSTR*, building on GaitMix and GaitRef [254], to integrate and refine skeletons with silhouettes for gait recognition. *GaitSTR* incorporates bone representation alongside the joints, emphasizing the numerical connectivities between different nodes. It combins silhouettes and skeletons with two levels of refinement: silhouette-to-skeleton refinement for general guidance and dual-layer cross-modal adapters for sequential two-stream refinement between the joints and bones, ensuring temporal consistency across different representations. We compare *GaitSTR* on four public datasets, including CASIA-B, OUMVLP, Gait3D, and GREW, and demonstrate state-of-the-art performance compare with other gait recognition methods.

Chapter 4

GaitHBS: Shape-Assisted Gait Recognition

4.1 Introduction

Gait recognition [71, 170, 213, 225] aims to find the uniqueness for a sequence of walking patterns and posture of a person in the binarized silhouette sequence describing human boundaries. However, appearance variances in 2-D images, like camera positions, carried-on objects, and clothing,introduce additional disparity in the human shape and make the task of recognition challenging, as shown in Figure 4.1 (a).

To address the issue of appearance variances, researchers have developed part-based deep-learning models that focus on local patterns. For example, GaitPart [44] splits the image into several patches to encode the part-based features for gait recognition, whereas GaitGL [106] utilizes local features along with the global ones for the analysis. By limiting the variances to a small portion of the feature, these strategies aim to reduce the influence of the variances. However, features encoded by these approaches are still impacted since they still focus on the 2-D domain.

In addition to 2-D gait representations, we propose inferring 3-D human body shape representations directly from silhouette sequences by using knowledge distillation to learn from a small number of RGB samples. We observe that the 3-D body shape of the human, as illustrated in Fig. 4.1 (b) for example, is, in principle, invariant to viewpoint, carrying objects, and clothing, and might therefore be useful for human identification under such challenging scenarios. Nevertheless, inference of the underlying 3-D shape is

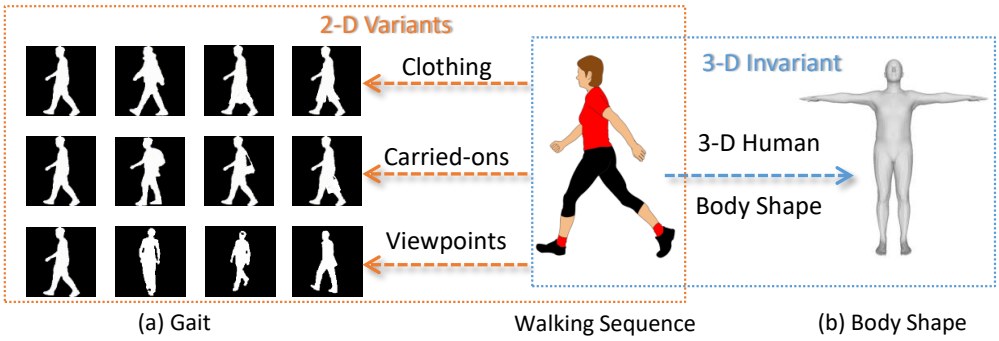


Figure 4.1: (a) 2-D silhouette sequences suffer from different appearance variances, such as clothing, carried-on bags and camera viewpoints. (b) However, 3-D skinned human body shape is robust and shows consistent output for the same person.

difficult in and of itself. Recent work [152, 139, 237] has created numerous approaches to infer 3-D shape from RGB images, but no work directly infers the body shape from a silhouette sequence. Many gait datasets lack RGB photos due to confidentiality, making such inferences more difficult.

To infer 3-D body shapes from the silhouette sequence, we exploit a temporal shift between the features obtained from adjacent frames in the silhouette sequence. Considering the consistency of the body shape of the same individual in a video sequence, we extract and reconstruct a single body shape for a video sequence. We combine body shapes acquired from our approach with 2-D gait features collected from certain state-of-the-art gait recognition methods [12, 44, 106, 76] to build our module on each of them. To supervise the generation of the body shape from the silhouette sequence, we distill and transfer the knowledge from a small set of RGB images, denoted as human body prior, and propagate it to gait. We demonstrate that adding 3-D body shape feature inferred from silhouette sequence significantly improves gait recognition accuracy on two public datasets (CASIA-B [226] and OUMVLP [179]), particularly for novel viewpoints that were not observed during training with fewer available training instances.

In summary, our contributions are summarized as follows: 1) We apply the 3-D human body inferred from gait to eliminate the effects of different clothing, carried-on objects and viewpoints for gait recognition; 2) We distill the knowledge of human body prior from limited single-frame RGB images and transfer

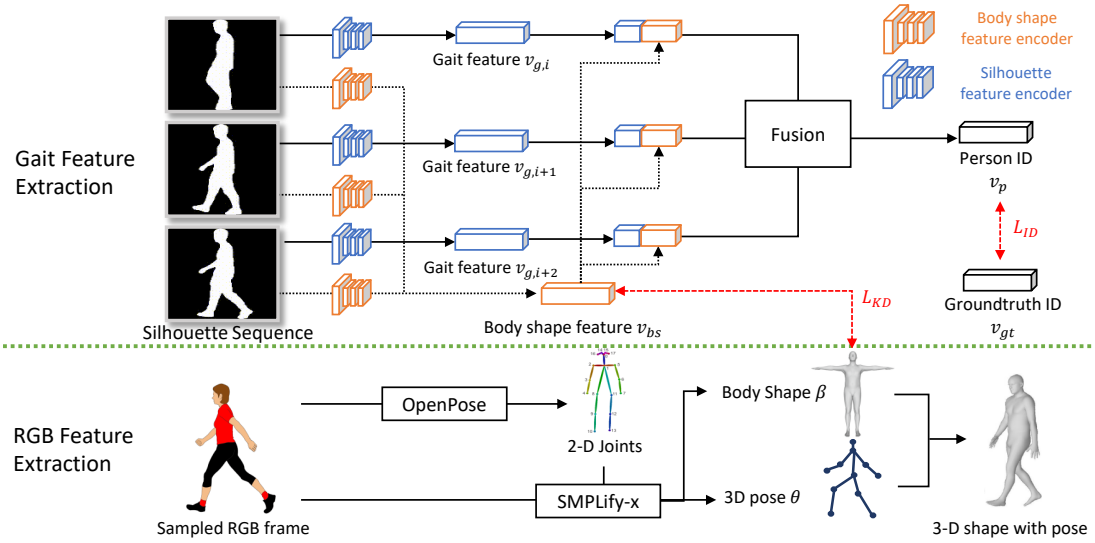


Figure 4.2: Our Proposed method for gait recognition with 3-D human body shape. During inference, only the features extracted from gait branch are used. Features from RGB images (below the green line) are only used for training when corresponding images are available.

to the silhouette sequence for body shape reconstruction directly from gait; and 3) We explore the setting for gait recognition on novel camera positions to assess the generalization of gait recognition models with fewer available data across different cameras for each identity in the gallery.

4.2 Method

Our network consists of two branches, one for gait feature extraction and the other for body shape feature extraction from RGB images, which is shown in Fig. 4.2. For gait inputs, we have a silhouette feature encoder and a body shape feature encoder to encode the gait and body shape feature. To supervise the generation of body shapes, we simultaneously extract knowledge from selected RGB frames using a human body reconstruction model. Then, we extract and transfer these inferred body shape information from RGB frames to the gait branch's body shape encoder.

In the remaining of this section, we will first introduce our gait pipeline in Sec. 4.2.1 for how gait features and body shape features are extracted for identification, and then discuss how body shape of selected RGB images are generated and used as the supervision for the gait branch in Sec. 4.2.2.

4.2.1 Gait Feature Extraction

We propose two feature encoders to extract features: a silhouette feature encoder to extract walking patterns from the gait sequence and a body shape feature encoder to extract the body shape.

Silhouette Feature Encoder. The silhouette feature encoder projects the individual frames $\{f_i\}$ of a gait sequence G to their feature representings $\{v_{g,i}\}_{i=1,\dots,m}$, where m is the number of frames. To verify the generality of using the body shape features to improve the gait recognition network, we choose four state-of-the-art gait recognition methods as the gait feature encoder for comparison: GaitSet [12], GaitPart [44], GaitGL [106], and GLN [76].

Body Shape Feature Encoder. To extract the body shape feature from the silhouette sequence, we input the gait sequence G and project it to the feature space v_{bs} representing the body shape of the person in the video. Extracting the body shape from a single gait sequence is difficult since the single binary silhouette only provides the boundary of a human body and lacks essential information. Thus we need the neighbor frames to help complete the missing information for extracting the whole human body shape. We show our proposed body shape feature encoder in Fig. 4.3. The encoder consists of n blocks, where every block includes a convolution and a temporal shifting (TS) operation. The convolution operator takes the frame-wise feature from the raw gait sequence or the previous layer as input, and operators in the same block share weight. After the convolution operation in each block, we follow [108] to exchange 12.5% of the features of part of the channels between the previous and future segments for temporal shifting.

We preserve the first frame of the sequence's content, which should be exchanged with the previous frame since there is no frame before it. We also keep the feature from the future segments for the last frame. After the last layer of feature shifting, we do a temporal average pooling on the extracted feature sequences to generate the final body shape feature v_{bs}. With the features from two encoders, we concatenate the body shape feature v_{bs} to each of the gait features $\{v_{g,i}\}_{i=1,\dots,m}$. We maxpool the features along the temporal and horizontal dimension following the implementation of [12, 44, 106, 76] and apply two fully-connected

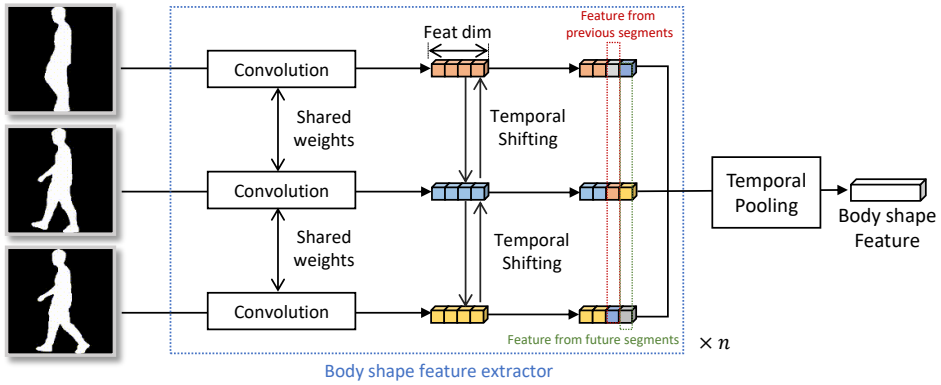


Figure 4.3: Our proposed body shape encoder for silhouette sequence input. n represents the repeated time for the operations in the block.

layers, whose dimensionalities match with the backbone network we used as the silhouette feature encoder, to generate feature v_p, representing the person's identity. We calculate the similarity between v_p and the groundtruth v_{gt} and calculate the identity loss L_{ID} following [12, 44, 76, 106].

4.2.2 Human Body Prior Distillation and Transfer

The purpose of inferring 3-D human body shapes is to separate movement patterns from variations in the appearance of 2-D silhouettes. Due to the absence of ways to directly reconstruct the 3-D human body from the gait, we first extract the shape priors from a small set of RGB frames in the gait sequence, then distill and transfer this information to the body shape feature encoder in the gait branch if corresponding RGB images are available.

Body Prior Inference. To infer 3-D body prior from the RGB images, we follow [139] for using SMPL-X reconstructed from SMPLify-X as body shapes. Compared with other 3-D body reconstruction models such as PIFu [152] and PIFuHD [153], SMPLify-X models human bodies with a strong human prior for the skinned body and outputs consistent results for the same person with different appearances, such as clothing, helping gait encoders to distinguish body shape from appearance variances in 2-D silhouettes. In addition, as a parametric method, SMPLify-X provides us the body shape feature decoupled from its pose and its strong prior can help us generate the complete shape even with some mild occlusions.

Considering the time consumption for inferring body prior with SMPLify-X [139], it is not feasible to extract the body prior for all frames in the video or image sequence. Since the identity of the person within the same video is consistent, the inferred 3-D body prior without the pose should be identical across all frames in which the person is discernible. Consequently, we infer the prior form based on one of the frames taken from the RGB sequence or video in conjunction with the gait sequence G.

To select this frame, we first extract the skeletons $\{s_i\}_{i=1,\ldots,m}$ of the whole sequence using OpenPose [7], followed by finding the longest sequence in the segments of $\{s_i\}$ with skeletons detectable and use the middle frame of this segment, which we annotate as s_r, to represent the body prior of the whole video. In this way, we can guarantee the quality of image s_r used to infer the 3-D body prior since the longest segments with skeletons detectable can ensure stable and consistent performance for pose detection and estimation, making it easier for body prior extraction.

We then reconstruct the whole human body prior for s_r by generating the shape feature $beta$ and 3-D pose θ following [139]. β is a 1-D vector with a size of 10 describing the appearance of the reconstructed body prior, and θ only includes 3-D joint positions. By combining the β and θ, we can reconstruct a full 3-D body model for a specific pose. In our experiments, we only use the β as the body prior feature v_{br} to guide the body shape based on silhouette v_{bs}. No skeleton information is used for gait recognition to avoid the different accuracies of the prediction of skeleton.

Knowledge Distillation and Transfer. With the body prior features v_{br} from the selected RGB frame and v_{bs} from the silhouette sequence, we utilize v_{br} to guide the training of feature v_{bs} from the body shape

feature encoder in the gait branch. We use CRD (Contrastive Representation Distillation) [182] for distilling knowledge between features from v_{br} and v_{bs} following

$$\begin{aligned} L_{KD} =& \mathbb{E}_{q(v_{br},v_{bs}|C=1)}[\log\ h(v_{br}, v_{bs})] \\ &+ \mathbb{E}_{q(v_{br},v_{bs}|C=0)}[\log(1 - h(v_{br}, v_{bs}))] \\ h(s,t) &= \frac{\exp\big(f_1(v_{bs})^T \cdot f_2(v_{br})\big)}{\exp(f_1(v_{bs})^T \cdot f_2(v_{br})) + \frac{N}{M}} \end{aligned} \tag{4.1}$$

where f_1 and f_2 are two linear projection layer with L_2 norm for projecting v_{br} and v_{bs}. N is the batch size and M is the cardinality of the dataset. C is 1 while v_{br} and v_{bs} are from the same identity and 0 if not. We will compare CRD with some other knowledge distillation methods in ablation studies. During training, we have two different loss functions, L_{ID} for gait recognition loss and L_{KD} for knowledge distillation between the features of inferred 3-D body prior from the selected RGB frame and the gait sequence, v_{br} and v_{bs}. We use a hyperparameter λ for balancing two losses. The final objective is shown as

$$L = \lambda_1 L_{ID} + \lambda_2 L_{KD} \tag{4.2}$$

We set λ_1 to 1 empirically and set λ_2 to 1 for knowledge transfer for the examples with RGB images and 0 for others.

4.3 Experiments

In this section, we show the details of our implementation for the experiment and the results.We first discuss our setups for the experiments in Sec. 4.3.1, followed by our analysis based on the experiment results in Sec. 4.3.2.

4.3.1 Experimental Setup

For experimental setup, we discuss datasets with the baseline methods and criteria. We also introduce the new setting of gait recognition where camera positions for training and test are mutually exclusive.

Datasets. We conduct our experiments on two public datasets, CASIA-B [226] and OU-MVLP [179].

- *CASIA-B* [226] is a gait recognition dataset with 124 objects with 10 different walking variants for each subject, where 6 are for normal walking (NM), 2 for walking while carrying bags (BG) and 2 for different clothing (CL). Each variant is recorded from 11 different camera viewpoints between 0° and 180° with 18 as the gap, making 110 videos for each subject. We follow [12, 44, 76, 106] to use silhouette sequences of the first 74 subjects for training. During inference, we use the first four walking variances in normal walking conditions (NM) as the gallery set, which is the identity library for the test set. The remaining 2 variants in NM, along with the sequences in BG and CL for the remaining 50 subjects, are used as probes for evaluation to find the correct identity in the gallery.

  In addition to using all the camera positions for supervised gait recognition, we introduce a new zero-shot setting where camera viewpoints used for training and testing are mutually exclusive. For all the camera viewpoints in the dataset, we only sample partial angles between 0° and 90° for training and use the viewpoints between 108° and 180° for inference to assess the model's performance when encountering novel viewpoints.

- *OUMVLP* [179] is a large gait recognition dataset with 10,307 subjects. Each subject has 2 different sequences for normal walking (NM) recorded from 14 different camera positions, resulting in 28 gait sequences for each subject. The camera viewpoints are evenly distributed from 0° to 90° and 180° and 270 with a 15-degree gap. Following [12, 44, 76, 106], we use the 5,153 subjects with an odd index between the 1-*st* and 10,305-*th* as training examples and the remaining 5,154 for inference, where the first sequence for each subject is the gallery set and the second as the probe.

Implementation Details. To extract the silhouette features, we follow the original settings of baseline methods [12, 44, 106, 76] for setting the hyperparameters for the model. For GaitPart [44], GaitSet [12] and GaitGL [106], we resize input gait sequence g to the size of 64×44. We use Adam optimizer with 1e-4 as the learning rate and 0.9 as the momentum. We set the margin in separate triplet loss as 0.2. Batch size is set to (8, 16) for CASIA-B, and (32, 16) for OUMVLP. We set the maximum iteration and weight decay following [12, 44, 106, 76]. For GLN [76], the initial gait sequence is sampled to 128×88. We use SGD with 0.1 as the initial learning rate and reduce it to $\frac{1}{10}$ three times during training. The learning rate is reduced every 10,000 steps for CASIA-B and every 50,000 steps for OUMVLP.

For the body shape encoder at the gait branch, we apply the temporal shifting modules to MobileNet-v2 [154] and set n to 6 following [108] with the same learning rates and hyperparameters as the silhouette feature extraction model. We use CRD as our knowledge distillation method.

To fuse the inferred body shape feature with the gait features from the silhouette feature encoder, we append the features before the last fully-connected layers for each backbone, since the features before the temporal or set pooling are the high-level feature representing the frame-level identity, and the inferred 3-D body shape representation can give additional guidance for identity encoding. For GaitPart, we append the 3-D body shape feature to all the part features to help each feature for a specific part understand the global body shape along with its local patterns. The input of fully-connected layers is set to the original size of the identity feature plus 10 (size of v_{bs}) for each model [12, 44, 106, 76] after feature concatenation.

RGB Data for Knowledge Distillation. For 3-D human body prior extraction, we use the latent feature β in SMPL-X [119] model and normalize features in training set to $(0, 0.1)$ gaussian distribution. To supervise the generation of body shape feature in the gait branch, we select 20% of sequences in the CASIA-B sequence for the data distillation and transfer. Since OUMVLP does not provide the RGB video sequences, we apply the body shape feature encoder for the gait branch pretrained on the CASIA-B subset and keep it frozen during training for feature extraction for all the examples in the OUMVLP dataset.

Details for Identity Loss L_{ID}. For the selection of identity loss function L_{ID}, we follow the implementation of each baseline method [12, 44, 76, 106]. For GaitSet-HBS, GaitPart-HBS and GLN-HBS, We use the triplet loss with its margin set to 0.2 as L_{ID}. For GaitGL-HBS, in addition to the triplet loss with the same margin, we use a cross-entropy loss for predicting the identity, which is represented as a one-hot vectors; weights for both losses are set to 1.

Baseline Methods. Since our method is an additional to the existing gait recognition methods, we compare with four state-of-the-art deep-learning gait recognition methods: GaitSet [12], GaitPart [44], GaitGL [106] and GLN [76]. We compare the baseline methods with and without inferred 3-D human body shape on both datasets. For ablation studies, we conduct our experiments on GaitGL [106] and GLN [76], since these are the two state-of-the-art methods for gait recognition. We exclude GaitView [9] and Gait3D [237] as they have extra supervision or additional input modality (framewise skeletons and body meshes) from RGB images. We also exclude earlier methods, such as [213, 166, 170], which are not state-of-the-art.

Inference and Metrics. We assess L_2 similarity between features extracted from examples from gallery and probe sets, excluding the identical-view cases. We calculate the top-1 accuracies for finding the response with the smallest L_2 distance among the examples in the gallery to each example in the probe.

4.3.2 Results and Analysis

In this subsection, we present the results and analysis on CASIA-B [226] and OUMVLP [179]. We further conduct ablations on CASIA-B for the selection of 3-D body shape features along with knowledge distillation and transfer.

Results on CASIA-B. We show the results for CASIA-B in Table 4.1. Methods ending with 'HBS', which is the abbriviation of **H**uman **B**ody **S**hape, are the ones with inferred 3-D human body features compared with baseline methods. In addition, we summarize the statistics for the performance on CASIA-B in Table 4.2, where we compare the models with and without features for the inferred human body.

Table 4.1: Gait recognition results on CASIA-B dataset, excluding identical-view cases.

Probe	Method	Camera Positions											Mean
		0	18	36	54	72	90	108	126	144	162	180	
NM #5-6	GaitSet [12]	91.1	98.0	**99.6**	97.8	95.4	93.8	95.7	97.5	98.1	97.0	88.2	95.6
	GaitPart [44]	94.0	98.7	99.3	98.8	94.8	92.6	96.4	98.3	99.0	97.4	91.2	96.4
	GLN [76]	93.8	98.5	99.2	98.0	95.2	92.9	95.4	98.5	99.0	99.2	91.9	96.5
	GaitGL [106]	95.3	97.9	99.0	97.8	96.1	95.3	97.2	**98.9**	99.4	98.8	**94.5**	97.3
	GaitSet-HBS	92.2	98.7	99.2	97.9	95.1	93.4	95.7	98.4	98.2	97.9	89.0	96.0
	GaitPart-HBS	93.2	**98.9**	99.4	**98.9**	95.1	91.9	96.5	98.8	**99.5**	98.4	91.7	96.6
	GLN-HBS	93.8	98.1	99.1	98.2	95.2	94.2	95.4	98.4	99.2	**99.4**	93.2	96.8
	GaitGL-HBS	**96.0**	98.3	99.2	97.8	**96.4**	**95.9**	**97.4**	98.7	99.2	98.7	**94.5**	**97.5**
BG #1-2	GaitSet [12]	87.0	93.8	94.6	92.9	88.2	83.0	86.6	92.6	95.7	92.9	83.4	90.1
	GaitPart [44]	89.5	94.5	95.3	93.5	88.5	83.9	89.0	93.6	96.0	94.1	85.3	91.2
	GLN [76]	92.2	95.6	96.7	94.3	91.8	87.8	91.4	95.1	96.3	95.7	87.2	93.1
	GaitGL [106]	**93.0**	95.7	97.0	**95.9**	93.3	**90.0**	91.9	96.8	97.5	96.9	90.7	94.4
	GaitSet-HBS	89.7	93.8	95.9	93.3	87.1	83.1	87.4	91.9	94.1	93.7	85.1	90.5
	GaitPart-HBS	90.1	93.6	95.7	94.4	89.9	85.8	89.9	94.0	96.0	92.7	86.4	91.7
	GLN-HBS	91.7	**96.6**	96.6	95.2	90.9	88.1	91.5	95.4	96.6	96.8	89.8	93.6
	GaitGL-HBS	**93.0**	96.0	**97.3**	**95.9**	**93.7**	89.5	**92.9**	**97.0**	**98.3**	**97.4**	**92.2**	**94.8**
CL #1-2	GaitSet [12]	71.0	82.6	84.0	80.0	71.7	69.1	72.1	76.7	78.5	77.2	63.4	75.1
	GaitPart [44]	72.5	82.8	86.0	82.2	79.5	71.0	77.7	80.8	82.9	81.4	67.7	78.6
	GLN [76]	78.5	90.4	90.3	85.1	80.2	75.8	78.1	81.8	80.9	83.2	**72.6**	81.5
	GaitGL [106]	71.7	**90.5**	**92.4**	89.4	**84.9**	**78.1**	83.1	**87.5**	89.1	83.9	67.4	83.5
	GaitSet-HBS	72.9	84.1	83.7	79.6	73.0	70.5	73.1	76.6	79.8	78.3	64.6	76.0
	GaitPart-HBS	75.9	84.8	86.5	84.6	77.4	74.4	78.6	82.4	83.5	80.5	67.6	79.7
	GLN-HBS	77.7	89.4	91.9	87.0	84.1	**78.1**	81.6	83.8	85.2	83.8	**72.6**	83.2
	GaitGL-HBS	**75.8**	**90.5**	92.3	**90.0**	84.0	77.9	**83.3**	87.3	**89.3**	**85.1**	69.8	**84.1**

Mean and STD values in Table 4.2 refer to the average and standard deviation values of performance for different viewpoints for the same model. We have the following observations:

1. **Better performance.** Table 4.2 shows that the models with inferred human body shapes outperform the original ones on all four baselines for all three splits. For most of the viewpoints shown, the best performances among all models also appear in the model with the inferred 3-D body shape. With the knowledge of the boundary of the skinned human body model, gait recognition models are capable of focusing on the motions instead of the appearances in 2-D silhouettes.

2. **Stability at different viewpoints.** In addition to the average performance for all camera viewpoints, we observe the standard deviations for the accuracies at different viewpoints reduce after using inferred human body shapes. Even for those models with no improvement on the mean value, e.g., GaitPart-HBS compared with GaitPart on the NM split, the standard deviation still reduces.

Table 4.2: Statistics analysis for supervised results on CASIA-B dataset, excluding identical-view cases. (↑) indicates that larger values show better performance, while (↓) indicates that lower values are better. Δ indicates the change between the method with and without HBS.

Probe	Stats	HBS	Method				Avg. Change
			GaitSet [12]	GaitPart [44]	GLN [76]	GaitGL [106]	
NM #5-6	Mean (↑)	✗	95.6	96.4	96.5	97.3	***+0.3***
		✓	96.0	96.6	96.8	97.5	
		Δ	*+0.3*	*+0.2*	*+0.3*	*+0.2*	
	STD. (↓)	✗	3.4	2.8	2.8	1.7	***-0.1***
		✓	3.2	3.0	2.4	1.6	
		Δ	*-0.2*	*+0.2*	*-0.4*	*-0.1*	
BG #1-2	Mean (↑)	✗	90.1	91.2	93.1	94.4	***+0.4***
		✓	90.5	91.7	93.6	94.8	
		Δ	*+0.4*	*+0.5*	*+0.5*	*+0.4*	
	STD. (↓)	✗	4.6	4.2	3.3	2.7	***-0.3***
		✓	4.2	3.5	3.2	2.7	
		Δ	*-0.4*	*-0.7*	*-0.1*	*0.0*	
CL #1-2	Mean (↑)	✗	75.1	78.6	81.5	83.5	***+1.1***
		✓	76.0	79.7	83.2	84.1	
		Δ	*+0.9*	*+1.1*	*+1.7*	*+0.6*	
	STD. (↓)	✗	6.2	5.8	5.5	8.0	***-0.4***
		✓	5.9	5.6	5.5	7.0	
		Δ	*-0.3*	*-0.2*	*0.0*	*-1.0*	

With the inferred 3-D body shape, consistent for all camera positions, models can show additional robustness to the camera viewpoints and have more stable performances.

3. **Different appearance variances.** BG and CL sets have higher average accuracy than NM, whose gait appearances are similar. In BG and CL sets, the silhouette sequence individual is carrying different bags or wearing different outfits, affecting the binarized silhouette. Focusing on appearance differences hurts the gait recognition model. Since inferred 3-D human body shapes are skinned models, they are stable and resilient to these fluctuations. Gait recognition models may detect the consistent body shapes and reduce non-human body content, exhibiting benefits.

Zero-shot Results for Novel Viewpoints. In addition to the results on existing viewpoints, we assess the model on the novel viewpoints on CASIA-B dataset in Table 4.3 with GaitGL and GLN, the

Table 4.3: Gait recognition results for novel camera viewpoints on CASIA-B dataset. Viewpoints used for the training and inference stages are mutually exclusive. Supervised results, where all viewpoints are available for training, are shown at the top of each set.

Probe	Method	Training Viewpionts						Test Viewpionts					Mean	Avg.
		0	18	36	54	72	90	108	126	144	162	180		Diff.
NM #5-6	GLN	(All Camera Positions)						95.4	98.5	99.0	99.2	91.9	96.8	*+0.3*
	GLN-HBS	(All Camera Positions)						95.4	98.4	99.2	99.4	93.1	97.1	
	GLN	✓	✓	✓	✓	✓	✓	82.3	91.8	95.8	89.3	78.0	87.4	*+1.3*
	GLN	✓		✓		✓		79.7	88.2	94.7	87.5	78.0	85.7	
	GLN	✓			✓			74.0	89.3	93.5	83.8	76.3	83.3	
	GLN-HBS	✓	✓	✓	✓	✓	✓	85.5	93.5	97.3	89.0	82.3	89.5	
	GLN-HBS	✓		✓		✓		82.0	88.5	93.5	91.8	77.5	86.7	
	GLN-HBS	✓			✓			77.2	89.5	94.5	83.2	76.5	84.2	
	GaitGL	(All Camera Positions)						97.2	98.9	99.4	98.8	94.5	97.8	*-0.1*
	GaitGL-HBS	(All Camera Positions)						97.4	98.7	99.2	98.7	94.5	97.7	
	GaitGL	✓	✓	✓	✓	✓	✓	84.5	93.3	95.8	92.3	75.0	88.2	*+1.4*
	GaitGL	✓		✓		✓		81.5	90.0	92.8	89.5	69.5	84.7	
	GaitGL	✓			✓			76.3	91.0	91.3	86.2	69.7	82.9	
	GaitGL-HBS	✓	✓	✓	✓	✓	✓	88.0	93.8	95.5	92.5	78.8	89.7	
	GaitGL-HBS	✓		✓		✓		82.8	90.5	93.0	89.7	71.7	85.6	
	GaitGL-HBS	✓			✓			79.0	92.0	91.7	88.5	71.5	84.6	
BG #1-2	GLN	(All Camera Positions)						91.4	95.1	96.3	95.7	87.2	93.1	*+0.9*
	GLN-HBS	(All Camera Positions)						91.5	95.4	96.6	96.8	89.8	94.0	
	GLN	✓	✓	✓	✓	✓	✓	72.0	83.0	87.3	80.1	75.0	79.5	*+1.3*
	GLN	✓		✓		✓		70.7	79.2	88.5	80.0	73.5	78.4	
	GLN	✓			✓			65.0	81.5	86.5	79.6	65.5	75.6	
	GLN-HBS	✓	✓	✓	✓	✓	✓	74.5	85.0	88.8	82.1	74.0	80.9	
	GLN-HBS	✓		✓		✓		73.2	82.0	88.3	86.4	72.7	80.5	
	GLN-HBS	✓			✓			69.5	81.5	86.5	77.3	65.2	76.0	
	GaitGL	(All Camera Positions)						91.9	96.8	97.5	96.9	90.7	94.8	*+0.8*
	GaitGL-HBS	(All Camera Positions)						92.9	97.0	98.3	97.4	92.2	95.6	
	GaitGL	✓	✓	✓	✓	✓	✓	74.3	83.8	90.0	88.1	69.3	81.1	*+1.6*
	GaitGL	✓		✓		✓		72.2	81.0	85.7	84.1	61.5	76.9	
	GaitGL	✓			✓			64.7	82.7	86.5	78.5	66.8	75.9	
	GaitGL-HBS	✓	✓	✓	✓	✓	✓	75.5	87.8	91.8	87.4	70.5	82.6	
	GaitGL-HBS	✓		✓		✓		70.2	81.2	89.3	84.6	66.3	78.3	
	GaitGL-HBS	✓			✓			70.8	83.2	87.2	80.8	67.0	77.8	
CL #1-2	GLN	(All Camera Positions)						78.1	81.8	80.9	83.2	72.6	79.3	*+2.1*
	GLN-HBS	(All Camera Positions)						81.6	83.8	85.2	83.8	72.6	81.4	
	GLN	✓	✓	✓	✓	✓	✓	57.3	60.0	67.0	56.0	46.3	57.3	*+2.3*
	GLN	✓		✓		✓		50.0	62.5	67.5	58.5	44.8	56.5	
	GLN	✓			✓			45.0	54.7	59.3	52.0	44.5	51.1	
	GLN-HBS	✓	✓	✓	✓	✓	✓	57.8	62.5	68.3	61.5	46.8	59.4	
	GLN-HBS	✓		✓		✓		54.8	62.5	66.5	62.7	44.3	58.2	
	GLN-HBS	✓			✓			47.5	58.0	64.0	55.3	45.5	54.1	
	GaitGL	(All Camera Positions)						83.1	87.5	89.1	83.9	67.4	82.2	*+0.8*
	GaitGL-HBS	(All Camera Positions)						83.3	87.3	89.3	85.1	69.8	83.0	
	GaitGL	✓	✓	✓	✓	✓	✓	58.8	68.5	73.3	66.8	44.0	62.3	*+1.8*
	GaitGL	✓		✓		✓		53.2	63.7	71.2	63.5	41.0	58.5	
	GaitGL	✓			✓			48.5	62.0	64.2	51.5	43.3	53.9	
	GaitGL-HBS	✓	✓	✓	✓	✓	✓	61.5	70.8	76.0	72.3	47.3	65.6	
	GaitGL-HBS	✓		✓		✓		59.8	63.5	71.8	60.5	44.0	59.9	
	GaitGL-HBS	✓			✓			49.0	62.3	69.8	51.0	41.3	54.7	

Table 4.4: Gait recognition results on OUMVLP dataset, excluding identical-view cases.

Method	Camera Positions														Mean
	0	15	30	45	60	75	90	180	195	210	225	240	255	270	
GEINet [166]	23.2	38.1	48.0	51.8	47.5	48.1	43.8	27.3	37.9	46.8	49.9	45.9	45.7	41.0	42.5
GaitSet [12]	79.2	87.7	89.9	90.1	87.9	88.6	87.7	81.7	86.4	89.0	89.2	87.2	87.7	86.2	87.0
GaitPart [44]	82.8	89.2	90.9	91.0	89.7	89.9	89.3	85.1	87.7	90.0	90.1	89.0	89.0	88.1	88.7
GaitGL [106]	84.2	89.8	91.3	**91.7**	90.8	**91.0**	90.4	88.1	88.2	**90.5**	**90.5**	89.5	**89.7**	88.8	89.6
GaitSet-HBS	79.0	87.9	90.4	90.6	88.4	89.2	88.4	82.3	87.1	89.6	89.6	87.7	88.4	86.9	87.5
GaitPart-HBS	82.4	89.1	91.1	91.3	89.8	90.2	89.7	84.8	88.0	90.3	90.3	89.2	89.4	88.4	88.9
GaitGL-HBS	**84.7**	**90.2**	**91.4**	**91.7**	**90.9**	**91.0**	**90.5**	**88.4**	**88.7**	**90.5**	**90.5**	**89.6**	89.6	**88.9**	**89.8**

two of the best performing baselines. Instead of using silhouette sequence from all the viewpoints for both training and inference, we only use part of the viewpoints for training, and viewpoints used for training and inference are mutually exclusive. We notice that when gait recognition models encounter novel viewpoints not seen before, using the inferred human body shape gives a consistent improvement compared with the baseline methods. Although these novel camera positions are unavailable during training, the consistency of the 3-D human body shape helps gait recognition models extract motion information from a new camera position for identification.

We further reduce the number of available viewpoints during training to assess the robustness of gait recognition models learning from fewer examples. With fewer viewpoints available in the training set, performances for all the methods are decreasing. However, GaitGL [106] and GLN [76] with inferred 3-D human body shape still show a consistent improvement compared to the model without body shapes, showing the 3-D body shape can give consistent guidance at different amounts of data.

Results on OUMVLP. We show the results for the OUMVLP dataset in Table 4.4. Since the OUMVLP dataset does not provide the original RGB frames, we apply the knowledge distillation model pretrained on the training set of CASIA-B to infer human body shape directly from the silhouette sequences. Compared to baseline methods, inferring 3-D body shape for gait recognition consistently outperforms original methods, showing good generalization ability and robustness of body shape feature encoders across different datasets. Examples in OUMVLP are all normal walking with fewer variations, which explains the limited improvement as NM sets for CASIA-B.

4.4 Conclusion

In this chapter, we propose the exploitation of inferring 3-D body shape from gait sequence to disentangle gait motion from appearance variances of 2-D images. In addition to the gait pattern analysis, we distill the 3-D body shape features from selected RGB frames and transfer them to gait sequences via feature exchanging between neighbor frames. We assess our method with four state-of-the-art gait recognition methods and show better results on two public datasets at both seen and novel camera viewpoints.

Chapter 5

ShARc: Shape and Appearance-Based Person Re-Identification

5.1 Introduction

Different from the gait recognition, the video of a person in the wild may not include such fine-grained walking patterns. Recognizing individuals in-the-wild [131] is a challenging yet valuable task for determining a person's identity from images or videos, playing a crucial role in many applications. Since face images may be unreliable or unavailable for individuals at a distance or from specific viewpoints, recognizing individuals via body images or videos becomes increasingly important. In this chapter, we focus on video-level appearance and body shapes to develop a robust identification system suitable for various distances and camera viewpoints, utilizing multiple videos as gallery samples. We specifically address different clothing and activities in generalized scenarios by comparing and combining shape and appearance-based methods for identifying the person in a video segment.

To identify individuals from their body, research primarily focuses on appearance [207, 200] and gait [12, 44, 106, 237, 256, 254]. Unlike facial features, which are relatively constant [30, 88, 36], body appearance can vary significantly due to changes in clothing, environment, and occlusions [27], as depicted in Figure 1.1. Gait analysis captures an individual's walking pattern and is less affected by environmental

changes or clothing. However, it requires a walking sequence that may not always be available. Additionally, varying environmental conditions pose challenges in feature registration and matching, making the prediction of human identity more sensitive to noisy samples in gallery videos.

We introduce ShARc, a method based on **SH**ape and **A**ppearance **R**e**C**ognition. Specifically, we employ a Pose and Shape Encoder (PSE) and an Aggregated Appearance Encoder (AAE) to project the input video into their corresponding embedding spaces. Leveraging body shapes with shape and motion representations [256], ShARc enables identification in diverse scenarios; a robust body prior [119] offers guidance under occlusion or variations in clothing. Alongside this, we introduce multi-level appearance features for both video-level and frame-level analysis. Importantly, these techniques show commendable performance even before combining with body shapes.

To extract the shape of a person in a sequence, we disentangle motion and poses by extracting skeletons, 3-D body shapes, and silhouettes from tracklets with our Pose and Shape Encoder (PSE). We utilize silhouettes and 3-D body shapes to represent individual frame shape patterns in 2-D and 3-D space, while employing sequential skeletons to represent motions. For the two different shape modalities, we first extract their frame-wise features and then combine them frame-by-frame using an attention mechanism for body shape feature extraction. Subsequently, we concatenate the pooled features with pose features encoded from skeletons for the final shape representation.

Parallel to body shape extraction, we also use an Aggregated Appearance Encoder (AAE) to extract features from appearances, preserving identification information from raw images. We obtain both frame-wise and video-level features and integrate them for dual-level understanding. For frame-level extraction, we introduce a novel flattening layer after averaging to extract more distinguishable information and reduce overfitting. At the video level, we employ spatial and temporal attention, as per [200], to focus on key areas for person distinction. This allows the model to concentrate on unique patterns in both frame and sequence, instead of fully relying on one of the sources.

After obtaining both shape and appearance features, we employ centroid feature averaging for gallery registration, using the mean features of the same ID rather than comparing to each gallery separately. This helps to mitigate variances in gallery examples with different clothing. We validate our approach on public datasets like CCVID [59], MEVID [27], and the recently-released BRIAR [24], showing state-of-the-art performance on all of them.

In summary, our contributions are as follows: 1) We introduce ShARc, a multimodal method for person identification in-the-wild using video samples, focusing on both shape and appearance; 2) We unveil a novel Pose and Shape Encoder (PSE) that captures dynamic motion and body shape features for more robust shape-based identification; 3) We deploy an Aggregated Appearance Encoder (AAE) that incorporates both frame-level and video-level features.

5.2 Method

5.2.1 Shape-based Person Recognition

For shape-based person recognition, we mitigate the influence of appearance by focusing on alternative representations, such as 3-D human body shape and silhouettes, to emphasize the individual's body shape, as well as skeletons to capture motion in pose. Although gait recognition is useful when walking segments are available, it offers limited distinguishable information in stationary videos when the person is not walking. Unlike existing gait recognition methods [12, 106, 256, 180], our shape-based approach compensates for the absence of gait by leveraging extra body shape priors. We first extract the corresponding modalities utilized in our model, which include 3-D body shapes, skeletons, and silhouettes, and then fuse them as the final representation.

Given a video with sequential frames $V = \{f_i\}_n$ containing n frames of the person, ShARc decomposes it into two branches: the body shape $\{b_i\}$ and the RGB appearance of the frames $\{a_i\}$ that exhibit the most

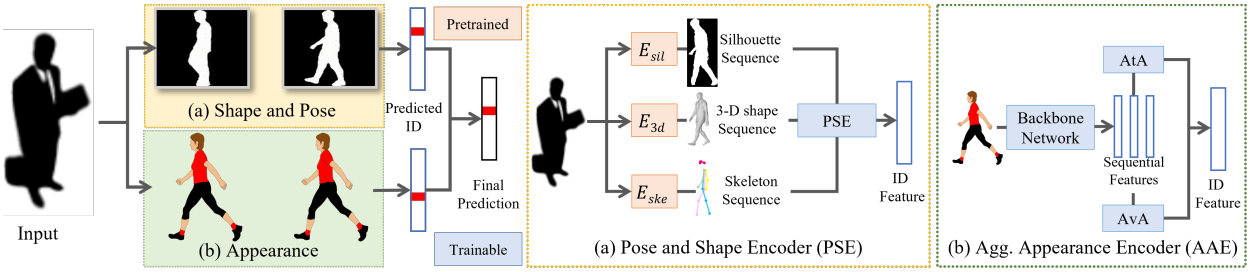


Figure 5.1: ShARc includes two sub modules: (a) a shape-based recognition system, PSE, which extracts the silhouette, 3-D body shape and skeletons sequences and fuses them for person recognition, and (b) an appearance-based recognition system, AAE, which takes both outputs from attention-based aggregation (AgA) and averaging aggregation (AvA) as input for identification.

distinguishable patterns, as illustrated in Figure 5.1. By estimating their independent similarities $S_{shape}(V)$ and $S_{app}(V)$ compared with gallery candidates, ShARc combines the two scores together using weighted average for the final similarity $S(V)$.

Shape and motion extraction. For shape-based person recognition, we focus on two crucial representations for distinguishing individuals: body shape P_i and motion M_i. Body shape encompasses specific actions or shapes a person may exhibit, while motion refers to the temporal information, representing a more specific case. If both shape and motion exist in all sequences, the task can be regarded as gait recognition. For body shape extraction, we focus on two distinct modalities: silhouettes and 3-D body shapes. Silhouettes represent the 2-D human boundary in each frame, while 3-D body shape reconstruction remains invariant to viewpoints by reconstructing the person's 3-D shape. The combination of silhouettes and body shapes allows for the preservation of both general shape and frame-wise detailed reconstruction of the individual.

In addition to body shape, we incorporate skeletons to understand motions, as motions represent the specific movement patterns of a person. Unlike gait recognition tasks [12], which use binarized silhouettes as input, skeletons can provide temporal understanding without the biases of body shape. Furthermore, by separating body shapes from motion analysis, the network for pose extraction can better focus on

the general shape, aiding temporal understanding and helping the model to maximize the utilization of potential information in the sequence.

For the three modalties described above, we employ three extractors, $E_{sil}(\cdot)$, $E_{3d}(\cdot)$ and $E_{ske}(\cdot)$, to encode the corresponding representations of these three modalities for each frame i following

$$P_i = E_{sil}(f_i) + E_{3d}(f_i); \;\; M_i = E_{ske}(f_i) \tag{5.1}$$

and extract the corresponding body shape P_i and motion M_i inputs for further processing. For silhouette input, we concatenate the silhouette and the cropped RGB images using silhouette as masks, as our input, since this can provide more separation of the human part in the body shape. Since these modals requires heavy training to ensure a stable performance, we use pretrained networks to extract these representations, which we discuss in Section 5.3.1.

Multimodal Fusion. With these three modalities, we introduce PSE for combining framewise body shape features P_i along with motion pattern M_i, as illustrated in Figure 5.2. For feature representation of silhouettes $Feat_{sil}$ and 3-D body shapes $Feat_{3d}$, we use corresponding encoders F_{pose} to project $E_{sil}(f_i)$ and $E_{3d}(f_i)$ into their embedding space. We then apply the 3-D spatial transformation network [237] with skip connection and implement horizontal pyramid pooling HPP [12] with B bins after the encoder output for each frame following

$$\begin{aligned} I_{sil},\ I_{3d} &= F_{pose}(E_{sil}(f_i),\ E_{3d}(f_i)) \\ I_{pose} &= (I_{sil} \cdot I_{3d}) + I_{sil} \\ I_{pose} &= HPP(I_{pose}) \end{aligned} \tag{5.2}$$

where I_k represents the feature for the modality k. For motion representation, we utilize a motion encoder F_{motion} to extract multi-level spatial and temporal skeleton information and use average pooling along

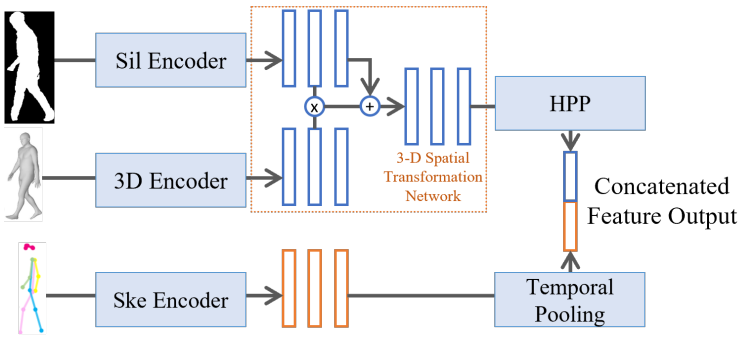


Figure 5.2: Architecture of PSE for combining body shape and motion information for shape-based identification.

the temporal dimension for the generated feature of the last layer. Then, we concatenate the skeleton feature, after temporal pooling, along with the pose representation as an additional new bin in the matching process, making the concatenated $(B+1) \times C$ feature map our final output for shape representation:

$$\begin{aligned} I_{motion} &= AvgPooling(F_{motion}(E_{ske}(f_i))) \\ I_{shape} &= [I_{pose}, I_{motion}] \end{aligned} \tag{5.3}$$

where $[\cdot, \cdot]$ represents feature concatenation.

5.2.2 Appearance-based Person Recognition

Compared to shape-based methods, which depend on the accuracy of body shape and contours, appearance provides richer and lossless RGB information for distinguishing individuals. We implement both attention-based and averaging appearance aggregation for identification. As people may wear different clothing and be in varying environmental conditions, we incorporate temporal and spatial information with attention-based appearance aggregation to focus on the relevant parts for differentiation between nearby frames. Moreover, to avoid overfitting on specific body parts or frames, we also employ video-level averaging aggregation to equally utilize spatial and temporal features.

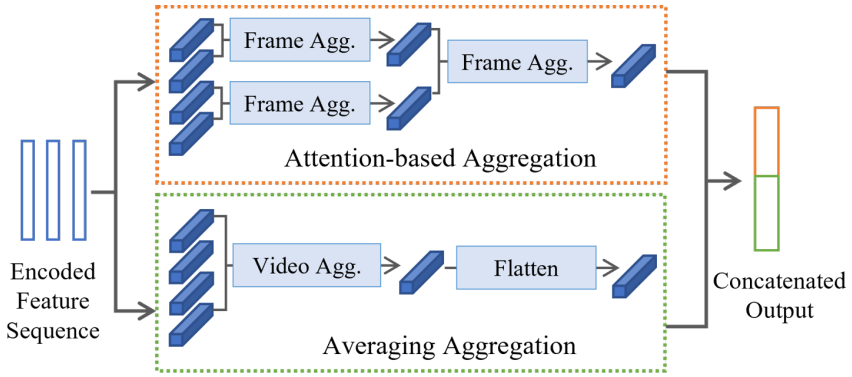


Figure 5.3: Architecture of the AAE with an example of sequence length $n = 4$. AAE aggregate the video frames in two ways: 1) attention-based aggregation, which mines the connection between nearby frames with attention, and 2) averaging aggregation, which takes all the frames together equally.

Attention-based Aggregation. For attention-based aggregation, we follow Figure 5.3 (a) for building spatial and temporal attention (STA) for the features extracted from the backbone network, encoding each frame F_i to their corresponding features A_i. We follow [200] to combine the features of two frames using a 3-level pyramid following

$$A_t^{l+1} = SA(A_t^l) + SA(A_{t+1}^l) + TA(A_t^l, A_{t+1}^l) \tag{5.4}$$

where l is the current layer in the pyramid, and t is the temporal stamp for the current frame. TA and SA are two attention generation layers following [200]. For each layer of the pyramid, we reduce the number of available appearance features to half the size of its previous layer, until we get the output feature representation in the last layer. This means the network, as an example, can handle at most 8 frames for the final feature $A_{attn}(V)$ with a three layers of pyramid. It is important to note that if attention-based aggregation is not combined with averaging aggregation and its backbone feature encoder not shared, it is degraded to the existing method PSTA [200] encoder.

Averaging Aggregation. As attention mechanism may create overfitting when there is shift between training and testing domain, we add averaging aggregation, as illustrated in Figure 5.3 (b), for global representation extraction. Video-level appearance focuses on finding the corresponding features of each frame

and treating all the frames equally. After extracting the framewise appearance feature A_i, we average the features of all the frames in the same video following

$$A_{avg} = \frac{1}{n}\sum_{i=1}^{n} A_i \tag{5.5}$$

We then use Gamma Correction γ in the range of $[0, 1]$ to flatten the features as a feature flatten layer following

$$A_{avg} = sgn(A_{avg}) \cdot ||A_{avg}||^{\gamma} \tag{5.6}$$

where $sgn(\cdot)$ is the sign function operated on channel-wise elements. Since the videos include multiple frames that may capture the person from different aspects, some of the specific representative features of this person may not be captured in all the frames. With $\gamma < 1$, the new feature are different from the old one in cases. When the feature value is close to the zero point (0), flattening layer makes the original value more distinguishable by increasing its absolute value. In addition, the flattening layer can also reduce the maximum value and avoid overfitting with feature values far from 0, making the network focus on more patterns instead of on just a few of them for making predictions.

5.2.3 Registration and Fusion

For person identification in the wild, it is essential to handle videos of individuals with varying clothing conditions, as gallery videos also exhibit differences in clothing, leading to variances in appearance. To address this issue, we follow [207] and construct a centroid representation for registering gallery examples. Assuming we have $k \times c$ features with a same ID, we average the k features and use the $1 \times c$ feature for representing this ID in the gallery. The averaging operation helps to mitigate the biases arising from

different clothing, as clothing across videos are assumed to be randomly distributed, while the appearance remains consistent.

Since shape and appearance are distinct modalities, we compute the features independently for each and match them with their corresponding modalities in the gallery candidates to obtain two matching scores, $S_{shape}(V)$ and $S_{app}(V)$. We then use a weighted average function to combine these two scores following

$$S(V) = \alpha S_{shape}(V) + (1 - \alpha) S_{app}(V), \tag{5.7}$$

where $S(V)$ is the final similarity score, $S_{shape}(V)$ and $S_{app}(V)$ are the shape-based and appearance-based similarity scores, respectively, and α is a weight parameter that balances the contributions of the two modalities. By adjusting α, we can find the optimal combination that leads to the best overall identification performance. Based on our ablation results in Section 5.3.2, we set α to 0.1 in our experiment.

5.2.4 Objectives

Considering that ShARc is a model for video-based identification, we train shape and appearance models separately, using end-to-end training for each. For the shape-based recognition model, PSE, we follow [237] and combine triplet loss $\mathcal{L}_{triplet}$ [158] with a margin of 0.2, along with cross-entropy loss $\mathcal{L}_{CE}$ as follows:

$$\mathcal{L}_{shape} = 0.1\, \mathcal{L}_{triplet} + \mathcal{L}_{CE} \tag{5.8}$$

For the appearance model, we apply four losses following [207], which combines a Triplet loss $\mathcal{L}_{triplet}$ [158] with 0.3 as margin, a Center Loss $\mathcal{L}_{cen}$ [205], a Cross Entropy loss $\mathcal{L}_{CE}$, and a Centroid Triplet Loss $\mathcal{L}_{CTL}$ [207], as follows:

$$\mathcal{L}_{app} = \mathcal{L}_{triplet} + \mathcal{L}_{CE} + \mathcal{L}_{cen} + 5e^{-4}\, \mathcal{L}_{CTL} \tag{5.9}$$

5.3 Experiments and Results

In this section, we first discuss our experiment settings, such as datasets selection and model details, in Section 5.3.1, followed by the results and discussion in Section 5.3.2.

5.3.1 Experimental Details

In this subsection, we first discuss the datasets used in our experiments, followed by implementation details, baseline methods, and evaluation metrics.

Datasets. In our experiment, we primarily compare our method with other state-of-the-art methods on three challenging, public, video-based datasets: CCVID [59], MEVID [27], and BRIAR [24]. We include the statistics for these three datasets in Table 5.1. CCVID [59] and MEVID [27] are recent datasets featuring the same and different clothes and include more than one outfit for each identity, with 226 and 158 identities, respectively. Unlike CCVID, which has only one viewpoint from the same location, MEVID includes 33 viewpoints and multiple scales of images from 33 different settings. BRIAR is a large, in-the-wild person identification dataset with varying distances, conditions, activities, and outfits for identification.

Compared to CCVID and MEVID, BRIAR [24] encompasses more variations of distances, viewpoints, and candidate IDs, which models the person identification problem in the wild. In addition, BRIAR has more distractor IDs in the gallery for the open-set problem evaluation, as well as featuring more images from elevated cameras and UAVs, introducing greater difficulty for final template matching. Since the BRIAR dataset is continuously expanding, we use the version including both BGC1 and 2 following [24], which is an extended version compared to [62].

Implementation Details. We first discuss the detailed architecture used for shape and appearance-based networks, followed by the training and inference details.

Table 5.1: Statistics for the three datasets in our experiment.

Dataset	Split	#frames	#identities	#tracklets
BRIAR	train	4,366,198	407	37,466
	query	189,819	192	886
	gallery	2,326,111	544	4,379
CCVID	train	116,799	75	948
	query	118,613	151	834
	gallery	112,421	151	1,074
MEVID	train	3,609,156	104	6,338
	query	205,044	52	316
	gallery	981,207	54	1,438

Shape-based Modalities Extraction. For shape-based recognition, our model requires three different inputs: silhouettes, 3-D body shapes, and skeletons. For silhouette extraction $E_{sil}(\cdot)$, we use DeepLab-v3 [16] with ResNet-101 [67] pretrained on the Pascal VOC dataset as the backbone to identify the pixels predicted as the 'person' category for silhouettes. For the 3-D human body shape extraction $E_{3d}(\cdot)$, we use ROMP [177] pretrained on Human3.6M [79] and MPI-INF-3DHP [124] to extract three vectors: a 3-D camera parameter, a 10-D vector body shape, and a 72-D vector representing the rotation of the joints. These three vectors form an 85-D SMPL [119, 253] representation for each frame. Since there is only one person in each frame sequence, we use the first SMPL body shape predicted by ROMP as our body shape representation. For skeletons $E_{ske}(\cdot)$, we follow [180] and use HRNet [175] with architecture 'pose_-hrnet_w32' and 384×288 as input size, which is pretrained on the MS COCO dataset [109] for 2-D pose estimation as the skeleton representation.

With different input modalities available for shape-based modal extraction, we use ResNet-9 [42] as the gait encoder, a 4-layer MLP [237] for 3-D body shape encoding, and MS-G3D [118] for skeleton encoding. All these three models are trained together with PSE end-to-end with the shape-based recognition model.

Appearance-based Recognition Model. For input frames f_i, we first employ a ResNet-50 [67] network which is pretrained on ImageNet [151] dataset for feature encoding to get their $H \times W \times C$ feature maps

Table 5.2: Identification results on BRIAR dataset.

Method	All Activities		Walking Sequences		Stationary Sequences	
	Rank 1	Rank 20	Rank 1	Rank 20	Rank 1	Rank 20
GaitSet [12]	15.3	40.5	27.7	64.5	7.3	24.9
GaitPart [44]	14.1	41.7	25.7	67.8	6.6	24.8
GaitGL [106]	15.6	45.1	28.0	67.2	7.5	30.8
GaitMix [254]	15.9	46.5	27.6	65.3	8.1	33.9
GaitRef [254]	17.7	50.2	29.9	69.4	9.5	37.2
SMPLGait [237]	18.8	51.9	25.2	63.4	14.6	44.3
PSE (Ours)	21.2	65.3	23.2	68.6	19.9	63.2
DME [62]	25.0	63.8	30.4	68.8	21.5	60.5
PSTA [200]	33.6	67.3	32.1	66.0	34.5	68.1
CAL [59]	34.9	71.4	34.7	71.0	35.0	71.7
TCL Net [74]	31.3	65.6	31.0	65.1	31.5	65.9
Attn-CL+rerank [138]	27.6	61.8	26.9	60.5	28.1	62.6
AAE (Ours)	38.3	81.8	37.6	79.0	39.5	83.7
ShARc	**41.1**	**83.0**	**39.4**	**80.7**	**42.2**	**84.5**

A_i before spatial pooling. For AAE, we follow [200] and use the patch level encoding for building a three-layer pyramid architecture with two different levels of attentions: temporal attention (TA) between two consecutive frames, and spatial attention (SA) of each frame. TA and SA of the same layer of the pyramid share weight, while those from different layers do not. The output attention is the same size as the input feature A_i, so we apply point-wise production for each input attention-feature pair and sum them up as the output, which is the input for the next level of the pyramid. For averaging aggregation, we set γ as 0, which degrades the function to a binarized representation, following our results for ablation studies in Sec. 5.3.1. After having the two features from AAE, we concatenate them to represent the appearance of the person.

Training and Inference. Due to the network's complexity, we do not combine shape and appearance during training but train them individually end-to-end with their own inputs. For the shape-based network, we use the Adam optimizer for 180,000 iterations and set the initial learning rate as $1e^{-3}$. The learning rate is decayed to $\frac{1}{10}$ three times at iterations 30,000, 90,000, and 150,000. For the appearance-based method,

we follow [200] and train the network for 500 epochs, using the Adam optimizer with an initial learning rate of $3.5e^{-4}$. We decay the learning rate by 0.3 at steps 70, 140, 210, 310, and 410 during training.

During inference, we follow [207] by using centroid representation when registering the features of gallery examples via averaging all the features with the same ID. If there are multiple single frames, as gallery examples in BRIAR, we first combine the frames for the same ID as a 'pseudo video' before sending it into the network for feature extraction. When querying an example with the gallery, we use the cosine distance to find the highest score in the gallery for shape score $S_{shape}(V)$ and Euclidean distance for appearance score $S_{app}(V)$ following existing gait recognition works [12]. If videos are shorter than 8 frames, we resample the frames until we have 8 frames for appearance feature extraction, and if the video is longer than 8 frames, we separate the video into several groups of 8 frames and average the results after extracting the features from all the groups.

Baseline Methods and Metrics. In our experiment, we compare our method with some state-of-the-art person-reID methods on different datasets. For MEVID [27], we compared with CAL [59], AGRL [211], BiCnet-TKS [73], TCLNet [74], PSTA [200], PiT [228], STMN [41], Attn-CL [138], Attn-CL+rerank [138], and AP3D [60] following the official results in MEVID [27]. For CCVID [59], we compared with CAL [59] following their original paper setting. For the comparison on BRIAR, we select some re-ID methods [200, 59, 74, 138] based on their performance on MEVID, as well as including some gait-based recognition methods [12, 44, 106, 62] for comparison. For evaluation metrics, we use rank accuracies and mAP (mean average precision) for evaluation on these datasets.

5.3.2 Results and Analysis

To compare with existing methods, we present the results for different baseline methods on the BRIAR, MEVID, and CCVID datasets in Tables 5.2, 5.3, and 5.4, respectively. In addition, we conduct some further

Table 5.3: Rank accuracy and mAP on MEVID dataset. Results for existing methods are from official MEVID [27] implementation.

Methods	mAP	Rank-1	Rank-5	Rank-10	Rank-20
BiCnet-TKS [73]	6.3	19.0	35.1	40.5	52.9
PiT [228]	13.6	34.2	55.4	63.3	70.6
STMN [41]	11.3	31.0	54.4	65.5	72.5
AP3D [60]	15.9	39.6	56.0	63.3	76.3
TCLNet [74]	23.0	48.1	60.1	69.0	76.3
PSTA [200]	21.2	46.2	60.8	69.6	77.8
AGRL [211]	19.1	48.4	62.7	70.6	77.9
Attn-CL [138]	18.6	42.1	56.0	63.6	73.1
Attn-CL+rerank [138]	25.9	46.5	59.8	64.6	71.8
CAL [59]	27.1	52.5	66.5	73.7	80.7
PSE	10.6	25.9	39.9	48.7	62.7
AAE	**29.6**	59.2	**70.3**	**77.2**	**83.2**
ShARc	**29.6**	**59.5**	**70.3**	**77.2**	82.9

ablation studies along with visualizations of the attention generated by the appearance branch for analysis of why the appearance model still works for clothes changing cases.

Results for person identification. As our main experiment, we have compared with all the three datasets with state-of-the-art methods in Table 5.2, 5.3 and 5.4 respectively. Note that all these three datasets are describing the clothes change settings in the re-ID task, which is more complex than the existing person re-ID tasks with same outfit. We have the following observations.

(i) Identification Performance. Our proposed method, ShARc, demonstrates significant performance improvements on all three datasets when compared to other state-of-the-art methods. For instance, on the BRIAR dataset, SHARc, after combining shape and appearance, achieves a 6.2% and 11.6% improvement in rank-1 and rank-20 accuracy, substantially outperforming other state-of-the-art methods. Moreover, on the other clothes-changing datasets, our method attains a 2.5% and 7.5% improvement in mAP and Rank-1 accuracy on MEVID [27], as well as a 4.6% and 8.0% improvement on CCVID [59]. Note that we follow [27] not using centroid averaging for gallery on MEVID. In addition, unlike BRIAR and CCVID, activities in

Table 5.4: Rank-1 accuracy and mAP on CCVID dataset. CC includes the videos specifically for clothes changing, while general include both same and different clothing.

Method	General		CC	
	Rank-1	mAP	Rank-1	mAP
GaitNet [170]	62.6	56.5	57.7	49.0
GaitSet [12]	81.9	73.2	71.0	62.1
PSE (Ours)	83.9	86.5	77.1	85.0
CAL-baseline [59]	78.3	75.4	77.3	73.9
CAL Triplet [59]	81.5	78.1	81.1	77.0
CAL [59]	82.6	81.3	81.7	79.6
AAE (Ours)	89.7	89.9	84.6	84.8
ShARc	**89.8**	**90.2**	**84.7**	**85.2**

MEVID do not include specific walking patterns, which results in a limited contribution from the PSE when combined with the appearance-based method, AAE.

Apart from the overall dataset results, we note that gait-based methods [12, 44, 106, 62] and appearance-based methods [200, 59, 74, 138] display different performance differences for the two types of activities, standing and walking. On the BRIAR dataset, gait-based methods [12, 44, 106] struggle with stationary sequences. Although DME [62]* demonstrates reasonable performance by incorporating masked RGB images into the gait branch, it still faces challenges when gait information is not available. In contrast, appearance-based methods exhibit slightly better performance with stationary videos compared to walking sequences, as stationary videos have less blurred boundaries due to reduced motion.

(ii) Shape and Appearance Analysis. Apart from comparing our method with existing methods, we also separate the shape and appearance models, PSE and AAE, to evaluate their individual contributions in ShARc. We present the results in Table 5.2, 5.3, and 5.4. Our appearance-based approach, AAE, demonstrates a substantial improvement over other appearance-based methods and achieves the best performance. This suggests that the averaging aggregation is indeed effective in providing supplementary

*The BRIAR dataset has included more subjects compared to the version used in DME, making it considerably more challenging.

Table 5.5: Rank-1 accuracy for different distances in BRIAR.

Distances	200m	400m	500m	1000m	UAV
PSE	38.5	38.2	35.7	5.3	25.9
AAE	60.6	56.3	51.2	10.5	30.7
ShARc	64.3	60.4	56.0	10.5	36.4

information not captured by attention-based methods, thus helping to alleviate the overfitting problem. Furthermore, our shape-based model, PSE, not only outperforms other gait-based methods but also shows relatively robust performance on stationary videos, indicating that the integration of body shape features allows the model to better understand and distinguish between individuals, particularly when gait is unavailable.

It is worth noting that on datasets involving clothes-changing scenarios, such as BRIAR where the outfits between gallery and query videos are strictly different, appearance-based methods consistently outperform shape-based methods, even when both gait and body shape information are available. As shown in Table 5.2, appearance-based methods continue to surpass gait and body shape-based methods under different clothing conditions. One possible explanation for this observation is that the process of generating body shape (SMPL) and gait (silhouettes) features directly from RGB frames introduces noise or increases information loss during the preprocessing stage. This results in a degradation of the extracted features' quality and their effectiveness in the re-identification task.

On the other hand, appearance-based methods can effectively leverage the rich information provided by RGB images to focus on relevant areas, even when the patterns of outfits differ between gallery and probe videos. This finding highlights the potential limitations of human-designed features, such as gait patterns or 3-D body shape, which despite being specifically and carefully designed for certain tasks, may still lead to information loss and underperform when compared to machine-designed features. In the final part of this section, we will present visualizations that further illustrate the effectiveness of our appearance-based method in handling clothes-changing scenarios.

Table 5.6: Ablation results for different components in ShARc. 'att' and 'avg' are attention-based and averaging aggregations.

Distances	Rank 1	Rank 5	Rank 20
PSE	21.2	44.9	65.3
w/o binarized sil.	8.7	20.7	40.1
w/o skeletons	19.7	35.6	63.4
w/o 3-D shape	8.7	20.1	37.6
AAE	38.3	63.7	81.8
w/o att.	29.1	51.3	68.9
w/o avg.	33.0	57.2	77.5
w/o centroid [207]	30.9	56.1	75.4

Table 5.7: Rank-1 accuracy for feature flattening for AvA.

Gamma	1	0.2	0.1	0
Rank 1	35.1	36.6	37.5	38.3

Ablation results. Since the BRIAR dataset provides valuable information, such as the exact distance at which images are captured and the impact of different types of activities in the sequences, we conduct ablation experiments on a sampled validation set derived from the training sequences to analyze the selection of weights when fusing the scores from the shape and appearance models.

Distances. We present the performance of our method across various distances in Table 5.5. We select five distance variations from the BRIAR dataset: 200 meters, 400 meters, 500 meters, 1000 meters, and video captured from UAV cameras. Generally, performance is better at shorter distances. However, we see a significant performance drop at 1000 meters, where the bodies in images are nearly indistinguishable. The results for UAV-captured images aren't as strong as those at 200 meters. This is due to the incomplete body images, as the UAV images are taken with the head occluding the whole body. The performance decline of PSE is less compared to AAE, showing its relative robustness in identification when occlusion is present.

Model Components Ablations. Our pipeline consists of multiple sub-modules, and we analyze the individual contribution of each component in both branches. For gait representation, we have two components:

Table 5.8: Rank-1 accuracy for the selection of α.

App.	0.95	0.9	0.8	0.7	0.6
Shape	0.05	0.1	0.2	0.3	0.4
Rank 1	91.1	91.4	91.0	90.2	88.4

masked RGB and binarized silhouettes. We investigate the contributions of binarized silhouette masks and masked RGB images independently. It is important to note that the masked RGB images in this case are resized to a smaller scale, similar to binarized silhouettes, to provide information about the separation of body parts rather than directly using appearance for training. To remove each component in the network, we zero out the corresponding input for analysis.

We show the results in Table 5.6. For the shape-based branch, masked RGB contributes the most, while 3-D body shape and binarized silhouettes contribute almost equally. Compared to other modalities, 3-D masked RGB images precisely provide more internal content for the gait branch, enabling the network to understand the boundary of different body parts and the movement of each part. For the appearance branch, we find that both aggregation contribute similarly to the final performance, and the combination of both yields the best results. Furthermore, using centroid averaging [207] when registering gallery examples also has a significant contribution to the final performance.

Feature Flattening. For the flattening layer in averaging aggregation, we analyze the different Gamma and their corresponding results in Table 5.7. When Gamma is 1, we have simple averaging across all the features. We observe that with higher gamma values, our performance improves, indicating that the results exhibit more discriminative patterns. When gamma is infinity, the final feature representation becomes binarized and yields the best performance.

Choice of α. To combine the two modalities, we construct a small validation set from the training data to analyze the weights between appearance and shape models, and present the results in Table 5.8. We find that when the weight is 0.9 for appearance and 0.1 for shape, the model achieves the best performance.

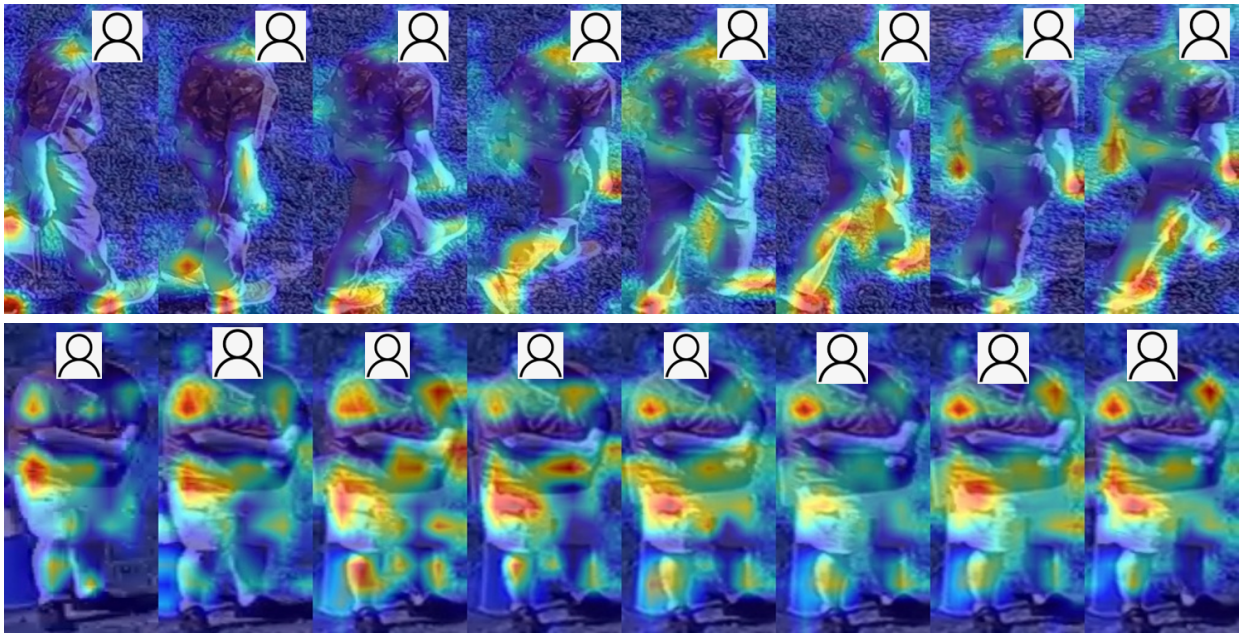


Figure 5.4: Attention generated from appearance model for (a) a walking sequence and (b) a stationary video for two examples taken from 100 meters distance category.

For shape-based methods, we use Euclidean distance instead of cosine distance; thus, 0.1 does not imply that it contributes minimally, but rather serves as a scaling factor for S_{shape} during combination. For generalizability, we use this γ and α for all datasets.

Visualization for Appearance Branch. In the BRIAR dataset, where query and gallery images feature distinct outfits, we use GradCam [160] to visualize network focus. Figure 5.4 presents two examples taken from 100-meter-distance cameras, one during walking and another while stationary. For walking videos, the network focuses mainly on the lower body and arms, suggesting implicit pose pattern extraction. In stationary scenarios, attention is directed towards the waist and shoulders, important areas for discerning body shape. We observe this trend across multiple examples, although quantification has not been performed.

5.4 Conclusion

In this chapter, we introduce ShARc, a shape and appearance-based method for identification in-the-wild. Our approach explicitly explores the contribution of body shape and appearance to the model with two encoders, pose and shape encoder for body shape and motion, and aggregated appearance encoder for human appearance. ShARc is able to handle most of the challenges for identification in the wild, such as occlusion, non-walking sequences, change of clothes, and image degradations. We have compared our method on three public datasets, including BRIAR, CCVID, and MEVID, and show state-of-the-art performance.

Chapter 6

Curriculum DeepSDF: Semantic Assistance for Object Construction

6.1 Introduction

In the last chapter, we introduce using SMPL [119] as the body shape representation to enhance the appearance and gait recognition for identifying the person in the video. SMPL is a skinned body model defined as a set of parameters for body shapes and poses, which cannot precisely model the overall body shape as the person. with a more fine-grained body shape representation, the overall recognition accuracy can also improve with more precise body shape reconstruction.

In recent years, 3D shape representation learning has aroused much attention [143, 145, 122, 57, 136]. Compared with images indexed by regular 2D grids, there has not been a single standard representation for 3D shapes in the literature. Existing 3D shape representations can be cast into several categories including: point-based [143, 145, 201, 37, 1, 148], voxel-based [212, 122, 144, 23], mesh-based [57, 61, 191, 168], and multi-view [171, 144, 184].

More recently, implicit function representations have gained an increasing amount of interest due to their high fidelity and efficiency. An implicit function depicts a shape through assigning a gauge value to each point in the object space [136, 125, 21, 126]. Typically, a negative, a positive or a zero gauge value represents that the corresponding point lies inside, outside or on the surface of the 3D shape. Hence, the shape is implicitly encoded by the iso-surface (e.g., zero-level-set) of the function, which can then

Figure 6.1: 3D reconstruction results of shapes with complex local details. From top to bottom: ground truth, DeepSDF [136], and Curriculum DeepSDF. We observe that the network benefits from the designed shape curriculum so as to better reconstruct local details. It is worth noting that the training data, training epochs and network architecture are the same for both methods.

be rendered by Marching Cubes [120] or similar methods. Implicit functions can also be considered as a shape-conditioned binary classifier whose decision boundary is the surface of the 3D shape. As each shape is represented by a continuous field, it can be evaluated at arbitrary resolution, irrespective of the resolution of the training data and limitations in the memory footprint.

One of the main challenges in implicit function learning lies in accurate reconstruction of shape surfaces, especially around complex or fine structure. Fig. 6.1 shows some 3D shape reconstruction results where we can observe that DeepSDF [136] fails to precisely reconstruct complex local details. Note that the implicit function is less smooth in these areas and hence difficult for the network to parameterize precisely. Furthermore, as the magnitudes of SDF values inside small parts are usually close to zero, a tiny mistake may lead to a wrong sign, resulting in inaccurate surface reconstruction.

Inspired by the works on curriculum learning [39, 5], we aim to address this problem in learning SDF by *starting small*: starting from easier geometry and gradually increasing the difficulty of learning. In this chapter, we propose a Curriculum DeepSDF method for shape representation learning. We design a

shape curriculum where we first teach the network using coarse shapes, and gradually move on to more complex geometry and fine structure once the network becomes more experienced. In particular, our shape curriculum is designed according to two criteria: *surface accuracy* and *sample difficulty*. We consider these two criteria both important and complementary to each other for shape representation learning: *surface accuracy* cares about the stringency in supervising with training loss, while *sample difficulty* focuses on the weights of hard training samples containing complex geometry.

Surface accuracy. We design a tolerance parameter ε that allows small errors in estimating the surfaces. Starting with a relatively large ε, the network aims for a smooth approximation, focusing on the global structure of the target shape and ignoring hard local details. Then, we gradually decrease ε to expose more shape details until $\varepsilon = 0$. We also use a shallow network to reconstruct coarse shapes at the beginning and then progressively add more layers to learn more accurate details.

Sample difficulty. Signs greatly matter in implicit function learning. The points with incorrect sign estimations lead to significant errors in shape reconstruction, suggesting that we treat these as hard samples during training. We gradually increase the weights of hard and semi-hard* training samples to make the network more and more focused on difficult local details.

One advantage of curriculum shape representation learning is that, it provides a training path for the network to start from coarse shapes and finally reach fine-grained geometries. At the beginning, it is substantially more stable for the network to reconstruct coarse surfaces with the complex details omitted. Then, we continuously ask for more accurate shapes which are relatively simple tasks, benefiting from the previous reconstruction results. Lastly, we focus on hard samples to obtain complete reconstruction with precise shape details. This training process can help avoid poor local minima as compared with learning to reconstruct the precise complex shapes directly. Fig. 6.1 shows that Curriculum DeepSDF obtains better

*Here, semi-hard samples are with the correct sign estimations but close to the boundary. In practice, we also decrease the weights of easy samples to avoid overshooting.

reconstruction accuracy than DeepSDF. Experimental results illustrate the effectiveness of the designed shape curriculum. Code will be available at https://github.com/haidongz-usc/Curriculum-DeepSDF.

In summary, the key contributions of this work are:

1. We design a shape curriculum for shape representation learning, starting from coarse shapes to complex details. The curriculum includes two aspects of *surface accuracy* and *sample difficulty*.

2. For surface accuracy, we introduce a tolerance parameter ε in the training objective to control the smoothness of the learned surfaces. We also progressively grow the network according to different training stages.

3. For sample difficulty, we define hard, semi-hard and easy training samples for SDF learning based on sign estimations. We re-weight the samples to make the network gradually focus more on hard local details.

6.2 Method

Our shape curriculum is designed based on DeepSDF [136], which is a popular implicit function based 3D shape representation learning method. In this section, we first review DeepSDF and then describe the proposed Curriculum DeepSDF approach. Finally, we introduce the implementation details.

6.2.1 Review of DeepSDF

DeepSDF is trained on a set of N shapes $\{X_i\}$, where K points $\{x_j\}$ are sampled around each shape X_i with the corresponding SDF values $\{s_j\}$ precomputed. This results in K (point, SDF value) pairs:

$$X_i := \{(x_j, s_j) : s_j = SDF^i(x_j)\}, \tag{6.1}$$

A deep neural network $f_\theta(z_i, x)$ is trained to approximate SDF values of points x, with an input latent code z_i representing the target shape.

The loss function given z_i, x_j and s_j is defined by the L_1-norm between the estimated and ground truth SDF values:

$$L(f_\theta(z_i, x_j), s_j) = |\text{clamp}_\delta(f_\theta(z_i, x_j)) - \text{clamp}_\delta(s_j)|, \tag{6.2}$$

where $\text{clamp}_\delta(s) := \min(\delta, \max(-\delta, s))$ uses a parameter δ to clamp an input value s. For simplicity, we use $\bar{s}$ to represent a clamping function with $\delta = 0.1$ in the rest of the paper.

DeepSDF also designs an auto-decoder structure to directly pair a latent code z_i with a target shape X_i without an encoder. Please refer to [136] for more details. At training time, z_i is randomly initialized from $\mathcal{N}(0, 0.01^2)$ and optimized along with the parameters θ of the network through back-propagation:

$$\arg\min_{\theta, z_i} \sum_{i=1}^{N} \left(\sum_{j=1}^{K} L(f_\theta(z_i, x_j), s_j) + \frac{1}{\sigma^2} ||z_i||_2^2 \right), \tag{6.3}$$

where $\sigma = 10^{-2}$ is the regularization parameter.

At inference time, an optimal z can be estimated with the network fixed:

$$\hat{z} = \arg\min_{z} \sum_{j=1}^{K} L(f_\theta(z, x_j), s_j) + \frac{1}{\sigma^2} ||z||_2^2. \tag{6.4}$$

6.2.2 Curriculum DeepSDF

Different from DeepSDF which trains the network with a fixed objective all the time, Curriculum SDF starts from learning smooth shape approximations and then gradually strives for more local details. We carefully design the curriculum from the following two aspects: *surface accuracy* and *sample difficulty*.

6.2.2.1 Surface accuracy.

A smoothed approximation for a target shape could capture the global shape structure without focusing too much on local details, and thus is a good starting point for the network to learn. With a changing smoothness level at different training stages, more and more local details can be exposed to improve the network. Such smoothed approximations could be generated by traditional geometry processing algorithms. However, the generation process is time-consuming, and it is also not clear whether such fixed algorithmic routines could meet the needs of network training. We address the problem from another view by introducing surface error tolerance ε which represents the upper bound of the allowed errors in the predicted SDF values. We observe that starting with relatively high surface error tolerance, the network tends to omit complex details and aims for a smooth shape approximation. Then, we gradually reduce the tolerance to expose more details.

More specifically, we allow small mistakes for the SDF estimation within the range of $[-\varepsilon, \varepsilon]$ for Curriculum DeepSDF. In other words, all the estimated SDF values whose errors are smaller than ε are considered correct without any punishment, and we can control the difficulty of the task by changing ε. Fig. 6.2 illustrates the physical meaning of the tolerance parameter ε. Compared with DeepSDF which aims to reconstruct the exact surface of the shape, Curriculum DeepSDF provides a tolerance zone with the thickness of 2ε, and the objective becomes to reconstruct any surface in the zone. At the beginning of network training, we set a relatively large ε which allows the network to learn general and smooth surfaces in a wide tolerance zone. Then, we gradually decrease ε to expose more details and finally set $\varepsilon = 0$ to predict the exact surface.

We can formulate the objective function with ε as follows:

$$L_\varepsilon(f_\theta(z_i, x_j), s_j) = \max\{|\bar{f}_\theta(z_i, x_j) - \bar{s}_j| - \varepsilon, 0\}, \tag{6.5}$$

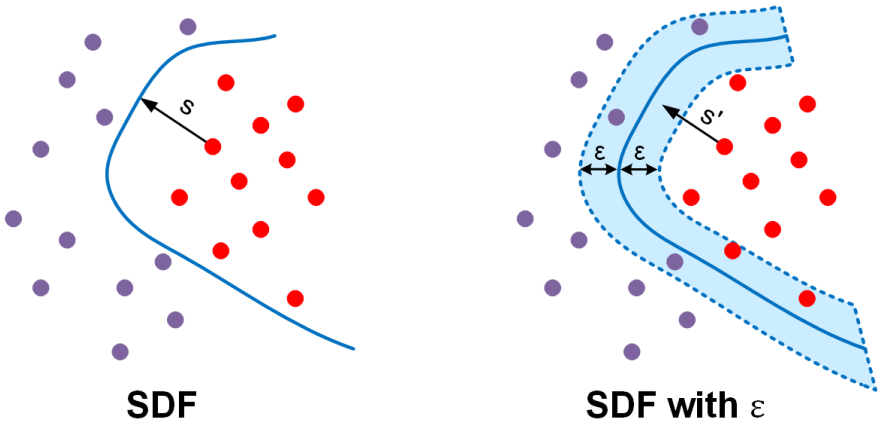


Figure 6.2: The comparison between original SDF and SDF with the tolerance parameter ε. With the tolerance parameter ε, all the surfaces inside the tolerance zone are considered correct. The training of Curriculum DeepSDF starts with a relative large ε and then gradually reduces it until $\varepsilon = 0$.

where (6.5) will degenerate to (6.2) if $\varepsilon = 0$.

Unlike most recent curriculum learning methods that rank training samples by difficulty [204, 64], our designed curriculum on shape accuracy directly modifies the training loss. It follows the formulation in [5] and also has a clear physical meaning for the task of SDF estimation. It is also relevant to label smoothing methods, where our curriculum has clear geometric meanings by gradually learning more precise shapes. We summarize the two advantages of the tolerance parameter based shape curriculum as follows:

1. We only need to change the hyperparameter ε to control the surface accuracy, instead of manually creating series of smooth shapes. The network automatically finds the surface that is easy to learn in the tolerance zone.

2. For any ε, the ground truth surface of the original shape is always an optimal solution of the objective, which has good optimization consistency.

In addition to controlling the surface accuracy by the tolerance parameter, we also use a shallow network to learn coarse shapes with a large ε, and gradually add more layers to improve the surface accuracy when ε decreases. This idea is mainly inspired by [87]. Fig. 6.3 shows the network architecture of the proposed Curriculum DeepSDF, where we employ the same network as DeepSDF for fair comparisons. After

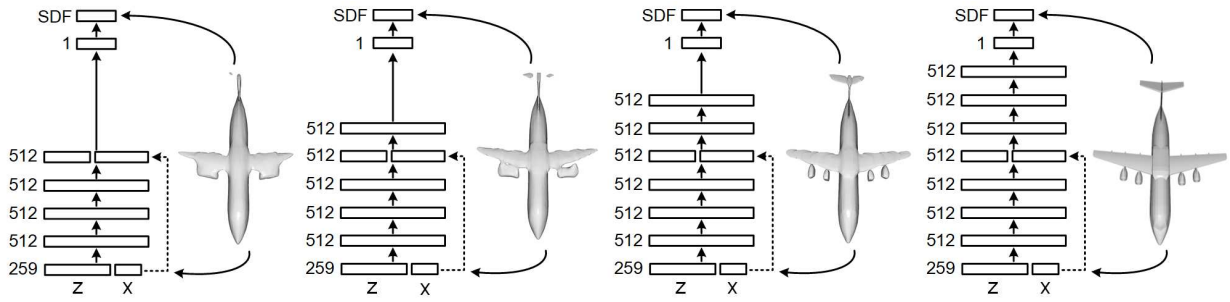


Figure 6.3: The network architecture of Curriculum DeepSDF. We apply the same final network architecture with DeepSDF for fair comparisons, which contains 8 fully connected layers followed by hyperbolic tangent non-linear activation to obtain SDF value. The input is the concatenation of latent vector z and 3D point x, which is also concatenated to the output of the fourth layer. When ε decreases during training, we add one more layer to learn more precise shape surface.

adding a new layer with random initialization to the network, the well-trained lower layers may suffer from sudden shocks if we directly train the new network in an end-to-end manner. Inspired by [87], we treat the new layer as a residual block with a weight of α, where the original link has a weight of $1 - \alpha$. We linearly increase α from 0 to 1, so that the new layer can be faded in the original network smoothly.

6.2.2.2 Sample difficulty.

In DeepSDF, the sampled points $\{x_j\}$ in X_i all share the same weights in training, which presumes that every point is equally important. However, this assumption may result in the following two problems for reconstructing complex local details:

1. Points depicting local details are usually undersampled, and they could be ignored by the network during training due to their small population. We take the second lamp in Fig. 6.1 as an example. The number of sampled points around the lamp rope is nearly 1/100 of all the sampled points, which is too small to affect the network training.

2. In these areas, the magnitudes of SDF values are small as the points are close to surfaces (e.g. points inside the lamp rope). Without careful emphasis, the network could easily predict the wrong signs.

Figure 6.4: Examples of hard, semi-hard and easy samples for (a) $s > 0$, and (b) $s < 0$. In the figure, s is the ground truth SDF, and we define the difficulty of each sample according to its estimation $f_\theta(z, x)$.

Followed by a surface reconstruction method like Marching Cubes, the wrong sign estimations will further lead to inaccurate surface reconstructions.

To address these issues, we weight the sampled points differently during training. An intuitive idea is to locate all the complex local parts at first, and then weight or sort the training samples according to some difficulty measurement [5, 204, 64]. However, it is difficult to detect complex regions and rank the difficulty of points exactly. We propose an adaptive difficulty measurement based upon the SDF estimation of each sample and re-weight the samples to gradually emphasize more on hard and semi-hard samples on the fly.

Most deep embedding learning methods judge the difficulty of samples according to the loss function [158, 35]. However, the L_1-norm loss can be very small for the points with wrong sign estimations. As signs play an important role in implicit representations, we directly define the hard and semi-hard samples based on their sign estimations. More specifically, we consider the points with wrong sign estimations as hard samples, with the estimated SDF values between zero and ground truth values as semi-hard samples, and the others as easy samples. Fig. 6.4 shows the examples. For the semi-hard samples, although currently they obtain correct sign estimations, they are still at high risk of becoming wrong as their predictions are closer to the boundary than the ground truth positions.

To increase the weights of both hard and semi-hard samples, and also decrease the weights of easy samples, we formulate the objective function as below:

$$L_{\varepsilon,\lambda}(f_\theta(z_i, x_j), s_j) = \left(1 + \lambda sgn(\bar{s}_j) sgn(\bar{s}_j - \bar{f}_\theta(z_i, x_j))\right) L_\varepsilon(f_\theta(z_i, x_j), s_j), \tag{6.6}$$

where $0 \leq \lambda < 1$ is a hyperparameter controlling the importance of the hard and semi-hard samples, $sgn(v) = 1$ if $v \geq 0$ and -1 otherwise.

The physical meaning of (6.6) is that we increase the weights of hard and semi-hard samples to $1 + \lambda$, and also decrease the weights of easy samples to $1 - \lambda$. Although we treat hard and semi-hard samples similarly, their properties are different due to the varying physical meanings as we will demonstrate in the experiments. Our hard sample mining strategy always targets at the weakness of the current network rather than using the predefined weights. Still, (6.6) will degenerate to (6.5) if we set $\lambda = 0$. Another understanding of (6.6) is that $sgn(\bar{s}_j)$ shows the ground truth sign while $sgn(\bar{s}_j - \bar{f}_\theta(z_i, x_j))$ indicates the direction of optimization. We increase the weights if this direction matches the ground truth sign and decrease the weights otherwise.

We also design a curriculum for sample difficulty by controlling λ at different training stages. At the beginning of training, we aim to teach the network global structures and allow small errors in shape geometry. To this end, we set a relatively small λ to make the network equally focused on all training samples. Then, we gradually increase λ to emphasize more on hard and semi-hard samples, which helps the network to address its weaknesses and reconstruct better local details. Strictly speaking, the curriculum of sample difficulty is slightly different from the formulation in [5], as it starts from the original task and gradually increases the difficulty to a harder objective. However, they share similar thoughts and the ablation study also shows the effectiveness of the designed curriculum.

6.2.3 Implementation Details.

In order to make fair comparisons, we applied the same training data, training epochs and network architecture as DeepSDF [136]. More specifically, we prepared the input samples X_i from each shape mesh which was normalized to a unit sphere. We sampled 500,000 points from each shape. The points were sampled more aggressively near the surface to capture more shape details. The learning rate for training

Table 6.1: The training details of our method. *Layer* shows the number of fully connected layers. *Residual* represents whether we use a residual block to add layers smoothly.

Epoch	0-200	200-400	400-600	600-800	800-1000	1000-1200	1200-2000
Layer	5	6	6	7	7	8	8
Residual	×	✓	×	✓	×	✓	×
ε	0.025	0.01	0.01	0.0025	0.0025	0	0
λ	0	0.1	0.1	0.2	0.2	0.5	0.5

the network was set as $N_b \times 10^{-5}$ where N_b is the batch size and 10^{-3} for the latent vectors. We trained the models for 2,000 epochs. Table 6.1 presents the training details, which will degenerate to DeepSDF if we train all the 8 fully connected layers by setting $\varepsilon = \lambda = 0$ from beginning to the end.

6.3 Experiment and Results

In this section, we perform a thorough comparison of our proposed Curriculum DeepSDF to DeepSDF along with comprehensive ablation studies for the shape reconstruction task on the ShapeNet dataset [10]. We use the missing part recovery task as an application to demonstrate the usage of our method.

Following [136], we report the standard distance metrics of mesh reconstruction including the mean and the median of Chamfer distance (CD), mean Earth Mover's distance (EMD) [150], and mean mesh accuracy [159]. For evaluating CD, we sample 30,000 points from mesh surfaces. For evaluating EMD, we follow [136] by sampling 500 points from mesh surfaces due to a high computation cost. For evaluating mesh accuracy, following [159, 136], we sample 1,000 points from mesh surfaces and compute the minimum distance d such that 90% of the points lie within d of the ground truth surface.

6.3.1 Shape Reconstruction

We conducted experiments on the ShapeNet dataset [10] for the shape reconstruction task. In the following, we will introduce quantitative results, ablation studies and visualization results.

Quantitative results. We compare our method to the state-of-the-art methods, including AtlasNet [57] and DeepSDF [136] in Table 6.2. We also include several variants of our own method for ablation studies. *Ours*, representing the proposed Curriculum DeepSDF method, performs a complete curriculum learning considering both surface accuracy and sample difficulty. As variants of our method, *ours-sur* and *ours-sur w/o* only employ the surface accuracy based curriculum learning with/without progressively growth of the network layers, where *ours-sur w/o* uses the fixed architecture with the deepest size; *ours-sam* only employs sample difficulty based curriculum learning. For a fair comparison, we evaluated all SDF-based methods following the same training and testing protocols as DeepSDF, including training/test split, the number of training epochs, and network architecture, etc. For AtlasNet-based methods, we directly report the numbers from [136]. Here are the three key observations from Table 6.2:

1. Compared to vanilla DeepSDF, curriculum learning on either surface accuracy or sample difficulty can lead to a significant performance gain. The best performance is achieved by simultaneously performing both curricula.

2. In general, the curriculum of sample difficulty helps more on lamp and plane as these categories suffer more from reconstructing slender or thin structures. The curriculum of surface accuracy is more effective for the categories of chair, sofa and table where shapes are more regular.

3. As we only sample 500 points for computing EMD, even the ground truth mesh has non-zero EMD to itself rising from the randomness in point sampling. Our performance is approaching the upper bound on plane and sofa.

Hard sample mining strategies. We conducted ablation studies for a more detailed analysis of different hard sample mining strategies on the lamp category due to its large variations and complex shape details. In the curriculum of sample difficulty, we gradually increase λ to make the network more and more focused on the hard samples. We compared it with the simple strategy by fixing a single λ. Table 6.3 shows

Table 6.2: **Reconstructing shapes from the ShapeNet test set.** Here we report shape reconstruction errors in term of several distance metrics on five ShapeNet classes. Note that we multiply CD by 10^3 and mesh accuracy by 10^1. The *average* column shows the average distance and the *relative* column shows the relative distance reduction compared to DeepSDF. For all metrics except for *relative*, the lower, the better.

CD, mean	lamp	plane	chair	sofa	table	average	relative
AtlasNet-Sph	2.381	0.188	0.752	0.445	0.725	0.730	-
AtlasNet-25	1.182	0.216	0.368	0.411	0.328	0.391	-
DeepSDF	0.776	0.143	0.243	0.117	0.424	0.319	-
Ours	**0.473**	**0.070**	**0.156**	**0.105**	**0.304**	**0.216**	**32.3%**
CD, median							
AtlasNet-Sph	2.180	0.079	0.511	0.330	0.389	0.490	-
AtlasNet-25	0.993	0.065	0.276	0.311	0.195	0.267	-
DeepSDF	0.178	0.061	0.098	0.081	0.052	0.078	-
Ours	**0.105**	**0.033**	**0.064**	**0.069**	**0.048**	**0.056**	**28.2%**
EMD, mean							
GT	0.034	0.026	0.041	0.044	0.041	0.039	-
AtlasNet-Sph	0.085	0.038	0.071	0.050	0.060	0.060	-
AtlasNet-25	0.062	0.041	0.064	0.063	0.073	0.064	-
DeepSDF	0.066	0.035	0.055	0.051	0.057	0.053	-
Ours-Sur w/o	0.057	0.032	**0.048**	0.046	0.049	0.046	13.2%
Ours-Sur	0.055	0.027	**0.048**	0.046	**0.048**	0.045	15.1%
Ours-Sam	0.055	0.027	0.053	0.050	0.051	0.048	9.4%
Ours	**0.052**	**0.026**	**0.048**	**0.044**	**0.048**	**0.044**	**17.0%**
Mesh acc, mean							
AtlasNet-Sph	0.540	0.130	0.330	0.170	0.320	0.290	-
AtlasNet-25	0.420	0.130	0.180	0.170	0.140	0.172	-
DeepSDF	0.155	0.044	0.104	0.041	0.120	0.097	-
Ours-Sur w/o	0.133	0.035	0.089	0.040	0.104	0.083	14.4%
Ours-Sur	0.121	0.034	0.082	0.039	0.098	0.078	19.6%
Ours-Sam	0.135	**0.031**	0.083	**0.036**	0.087	0.074	23.7%
Ours	**0.103**	**0.031**	**0.080**	**0.036**	**0.087**	**0.071**	**26.8%**

that the performance improves as λ increases until reaching a sweet spot, after which further increasing λ could hurt the performance. The best result is achieved by our method which gradually increases λ as it encourages the network to focus more and more on hard details.

For hard sample mining, we increase the weights of hard and semi-hard samples to $1 + \lambda$ and also decrease the weights of easy samples to $1 - \lambda$. As various similar strategies can be used, we demonstrate the effectiveness of our design in Table 6.4. We observe that both increasing the weights of semi-hard

Table 6.3: Experimental comparisons with using fixed λ for hard sample mining. The method degenerates to *ours-sur* when $\lambda = 0$. CD is multiplied by 10^3.

λ	0	0.05	0.10	0.25	0.50	0.75	Ours
CD, mean	0.639	0.606	0.549	0.538	0.508	0.567	**0.473**

Table 6.4: Experimental comparisons of different hard sample mining strategies. In the table, *H*, *S* and *E* are the hard, semi-hard and easy samples, respectively. For the symbols, $\uparrow$ is to increase the weights to $1 + \lambda$, $\downarrow$ is to decrease the weights to $1 - \lambda$ and - is to maintain the weights. *H(↑)S(↑)E(↓)* is the sampling strategy used in our method, while *H(-)S(-)E(-)* degenerates to *ours-sur*. CD is multiplied by 10^3.

Strategy	H(-)S(-)E(-)	H(-)S(-)E(↓)	H(-)S(↑)E(-)	H(-)S(↑)E(↓)
CD, mean	0.639	0.563	0.587	0.508
Strategy	H(↑)S(-)E(-)	H(↑)S(-)E(↓)	H(↑)S(↑)E(-)	H(↑)S(↑)E(↓)
CD, mean	0.676	0.661	0.512	**0.473**

samples and decreasing the weights of easy samples can boost the performance. However, it is risky to only increase weights for hard samples excluding semi-hard ones in which case the performance drops. One possible reason is that focusing too much on hard samples may lead to more wrong sign estimations for the semi-hard ones as they are close to the boundary. Hence, it is necessary to increase the weights of semi-hard samples as well to maintain their correct sign estimations. The best performance is achieved by simultaneously increasing the weights of hard and semi-hard samples and decreasing the weights of easy ones.

Table 6.5: Comparison of mean of EMD on the lamp category of the ShapeNet dataset with varying numbers of sampled points.

Number of points	500	2000	5000	10000
GT	0.034	0.008	0.008	0.004
DeepSDF	0.066	0.056	0.052	0.051
Ours-Sur w/o	0.057	0.053	0.050	0.048
Ours-Sur	0.055	0.052	0.049	0.048
Ours-Sam	0.055	0.053	0.050	0.048
Ours	**0.052**	**0.051**	**0.047**	**0.046**

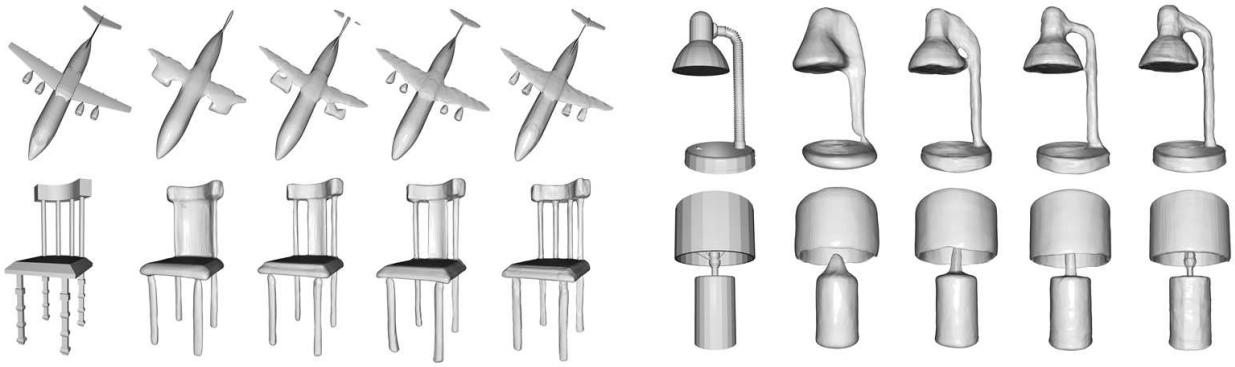


Figure 6.5: The visualization of shape reconstruction at the end of each training stage. From left to right: ground truth, 200 epochs, 600 epochs, 1000 epochs, and 2000 epochs.

Number of points for EMD. In Table 6.2, we followed [136] by sampling 500 points to compute accurate EMD, which would lead to relatively large distance even for ground truth meshes. To this end, we increase the number of sampled points during EMD computation and tested the performance on lamps. Results in Table 6.5 show that the number of sampled points can affect EMD due to the randomness in sampling, and the EMD of resampled ground truth decreases when using more points. Our method continuously obtains better results.

Visualization results. We visualize the shape reconstruction results in Fig. 6.1 to qualitatively compare DeepSDF and Curriculum DeepSDF. We observe that Curriculum DeepSDF reconstructs more accurate shape surfaces. The curriculum of surface accuracy helps to better capture the general structure, and sample difficulty encourages the recovery of complex local details. We also provide the reconstructed shapes at key epochs in Fig. 6.5. Curriculum DeepSDF learns coarse shapes at early stages which omits complex details. Then, it gradually refines local parts based on the learned coarse shapes. This training procedure improves the performance of the learned shape representation.

Table 6.6: Experimental comparisons under different ratios of removed points. CD and mesh accuracy are multiplied by 10^3 and 10^1, respectively.

Method	5%		10%		15%		20%		25%	
\Metric	CD	Mesh	CD	Mesh	CD	Mesh	CD	Mesh	CD	Mesh
plane										
DeepSDF	0.163	0.056	0.229	0.066	0.217	0.067	0.224	0.069	0.233	0.080
Ours	**0.095**	**0.032**	**0.124**	**0.044**	**0.149**	**0.052**	**0.163**	**0.062**	**0.192**	**0.072**
sofa										
DeepSDF	0.133	0.045	0.137	0.047	0.149	0.050	0.169	0.058	**0.196**	0.066
Ours	**0.110**	**0.037**	**0.120**	**0.041**	**0.143**	**0.046**	**0.165**	**0.053**	**0.196**	**0.061**
lamp										
DeepSDF	2.08	0.230	3.10	0.241	3.50	0.286	4.18	0.307	4.79	0.331
Ours	**1.96**	**0.167**	**2.87**	**0.195**	**3.27**	**0.231**	**3.52**	**0.277**	**4.07**	**0.320**

6.3.2 Missing Part Recovery

One of the main advantages of the DeepSDF framework is that we can optimize a shape code based upon a partial shape observation, and then render the complete shape through the learned network. In this subsection, we compare DeepSDF with Curriculum DeepSDF on the task of missing part recovery.

To create partial shapes with missing parts, we remove a subset of points from each shape X_i. As random point removal may still preserve the holistic structures, we remove all the points in a local area to create missing parts. More specifically, we randomly select a point from the shape and then remove a certain quantity of its nearest neighbor points including itself, so that all the points within a local range can be removed. We conducted the experiments on three ShapeNet categories: plane, sofa and lamp. In these categories, plane and sofa have more regular and symmetric structures, while lamp is more complex and contains large variations. Table 6.6 shows that part removal largely affects the performance on the lamp category compared with plane and sofa, and Curriculum DeepSDF continuously obtains better results than DeepSDF under different ratios of removed points. A visual comparison is provided in Fig. 6.6.

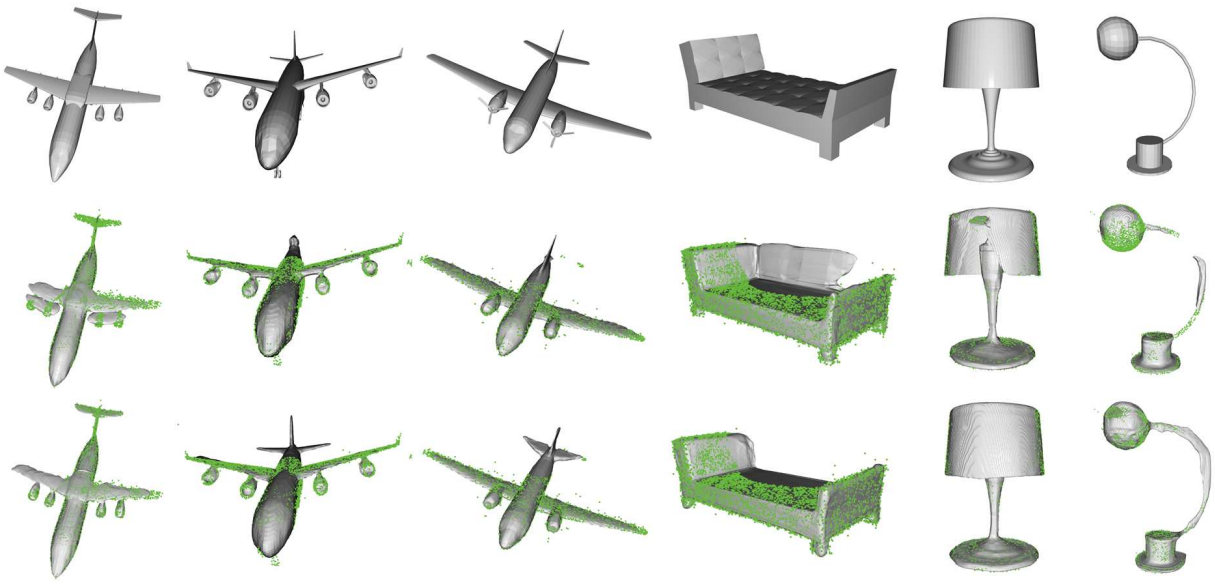


Figure 6.6: The visualization results of missing part recovery. The green points are the remaining points that we use to recover the whole mesh. From top to bottom: ground truth, DeepSDF, and Curriculum DeepSDF.

6.4 Conclusion

In this chapter, we have proposed Curriculum DeepSDF by designing a shape curriculum for shape representation learning. Inspired by the learning principle of humans, we organize the learning task into a series of difficulty levels from surface accuracy and sample difficulty. For surface accuracy, we design a tolerance parameter to control the global smoothness, which gradually increases the accuracy of the learned shape with more layers. For sample difficulty, we define hard, semi-hard and easy training samples in SDF learning, and gradually re-weight the samples to focus more and more on difficult local details. Experimental results show that our method largely improves the performance of DeepSDF with the same training data, training epochs and network architecture.

Chapter 7

CaesarNeRF: Few-views Generalizable NeRF

7.1 Introduction

Implicit representation is able to provide an object-level representation for reconstruction that assists downstream vision tasks. However, when the representation is limited to object level, it focuses on the overall shape reconstruction and discards high-frequency details. Such details are crucial to identify the person, as it can help distinguish across different identities. One of the solution to pay attention to these details is to calculate the loss based on the reconstructed 2-D images instead of the 3-D representation, since the minor changes in the surface of the object will be exaggerated in the projected view. Motivated by this, we investigate the possibility of NeRF [128] in this section.

Rendering a scene from a novel camera position is essential in view synthesis [6, 28, 189]. The recent advancement of Neural Radiance Field (NeRF) [129] has shown impressive results in creating photo-realistic images from novel viewpoints. However, conventional NeRF methods are either typically scene-specific, necessitating retraining for novel scenes [129, 208, 220, 46, 45], or require a large number of reference views as input for generalizing to novel scenarios [224, 13, 173, 187]. These constraints highlight the complexity of the few-shot generalizable neural rendering, which aims to render unseen scenes from novel viewpoints with a limited number of reference images.

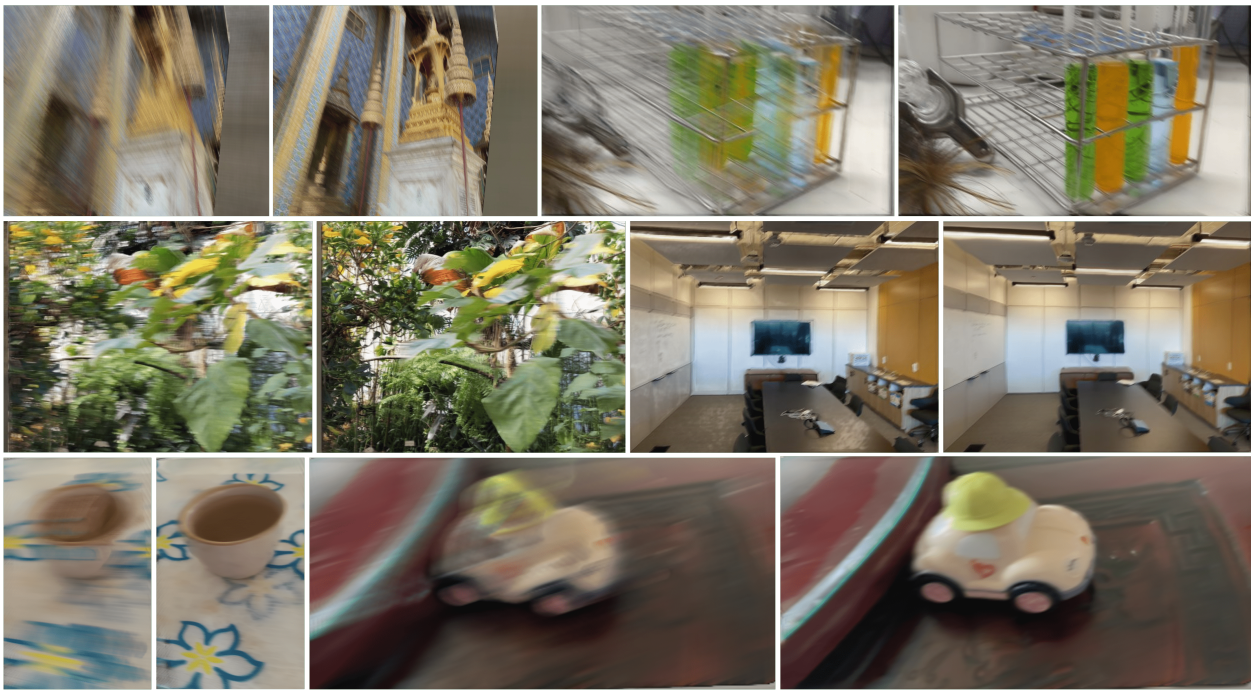


Figure 7.1: Novel view synthesis for novel scenes using **ONE** reference view on Shiny [208], LLFF [127], and MVImgNet [227] (top to bottom). Each pair of images corresponds to the results from GNT [187] (left) and CaesarNeRF (right).

Generalizing NeRF to novel scenes often involves using pixel-level feature embeddings encoded from input images, as seen in existing methods [224, 183]. These methods adapt NeRF to novel scenes by separating the scene representation from the model through an image encoder. However, relying solely on pixel-level features has its drawbacks: it requires highly precise epipolar geometry and often overlooks occlusion in complex scenes. Moreover, employing pixel-level features ignores the inherent interconnections within objects in the scene, treating the prediction of each pixel independently. This becomes problematic with a limited number of input reference images, as the data scarcity amplifies prediction ambiguity, significantly influenced by the biases of the input camera views.

We present CaesarNeRF, a method that advances the generalizability of NeRF by incorporating calibrated semantic representation. This enables rendering from novel viewpoints using as few as one input reference view, as depicted in Figure 7.1. Our approach combines semantic scene-level representation

with per-pixel features, enhancing consistency across different views of the same scene. The encoder-generated scene-level representations capture both semantic features and biases linked to specific camera poses. When reference views are limited, these biases can introduce uncertainty in the rendered images. To counter this, CaesarNeRF integrates camera pose transformations into the semantic representation, hence the term *calibrated*. By isolating pose-specific information from the scene-level representation, our model harmonizes features across input views, mitigating view-specific biases and, in turn, reducing ambiguity. In addition, CaesarNeRF introduces a sequential refinement process, which equips the model with varying levels of detail needed to enhance the semantic features. Extensive experiments on datasets such as LLFF [127], Shiny [208], mip-NeRF 360 [4], and the newly released MVImgNet [227] demonstrate that CaesarNeRF outperforms current state-of-the-art methods, proving effective in generalizable settings with as few as one reference view.

In summary, our contributions are as follows:

- We introduce CaesarNeRF, which utilizes scene-level calibrated semantic representation to achieve few-shot, generalizable neural rendering. This innovation leads to coherent and high-quality renderings.
- We integrate semantic scene context with pixel-level details, in contrast to existing methods that rely solely on pixel-level features. We also address view-specific biases by modeling camera pose transformations and enhance the scene understanding through the sequential refinement of semantic features.
- We demonstrate through extensive experiments that CaesarNeRF consistently outperforms state-of-the-art generalizable NeRF methods across a variety of datasets. Furthermore, integrating the Caesar pipeline into other baseline methods leads to consistent performance gains, highlighting its effectiveness and adaptability.

7.2 The proposed method

We first outline the general framework of existing generalizable NeRF in Section 7.2.1. Then, we present our proposed CaesarNeRF, as illustrated in Figure 5.1. This model integrates elements of semantic representation, calibration, and sequential refinement, detailed in Section 7.2.2, 7.2.3, and 7.2.4, respectively. The training objective is given in Section 7.2.5.

7.2.1 NeRF and generalizable NeRF

Neural Radiance Field (NeRF) [129, 128] aims to render 3D scenes by predicting both the density and RGB values at points where light rays intersect the radiance field. For a query point $\boldsymbol{x} \in \mathbb{R}^3$ and a viewing direction $\boldsymbol{d}$ on the unit sphere $\mathbb{S}^2$ in 3D space, the NeRF model $\boldsymbol{\mathcal{F}}$ is defined as:

$$\sigma, \boldsymbol{c} = \boldsymbol{\mathcal{F}}(\boldsymbol{x}, \boldsymbol{d}). \tag{7.1}$$

Here, $\sigma \in \mathbb{R}$ and $\boldsymbol{c} \in \mathbb{R}^3$ denote the density and the RGB values, respectively. After computing these values for a collection of discretized points along each ray, volume rendering techniques are employed to calculate the final RGB values for each pixel, thus reconstructing the image.

However, traditional NeRF models $\boldsymbol{\mathcal{F}}$ are limited by their requirement for scene-specific training, making it unsuitable for generalizing to novel scenes. To overcome this, generalizable NeRF models, denoted by $\boldsymbol{\mathcal{F}_G}$, are designed to render images of novel scenes without per-scene training. Given N reference images $\{\boldsymbol{I}_n\}_{n=1}^N$, an encoder-based generalizable NeRF model $\boldsymbol{\mathcal{F}_G}$ decouples the object representation from the original NeRF by using an encoder to extract per-pixel feature maps $\{\boldsymbol{F}_n\}_{n=1}^N$ from the input images. To synthesize a pixel associated with a point $\boldsymbol{x}$ along a ray in direction $\boldsymbol{d}$, it projects $\{\boldsymbol{F}_n\}_{n=1}^N$

from nearby views and aggregates this multi-view pixel-level information using techniques such as average pooling [224] or cost volumes [13]. This results in a fused feature embedding $\widetilde{\boldsymbol{F}}$, allowing $\boldsymbol{\mathcal{F}_G}$ to predict density σ and RGB values $\boldsymbol{c}$ for each point along the ray, as expressed by:

$$\sigma, \boldsymbol{c} = \boldsymbol{\mathcal{F}_G}(\boldsymbol{x}, \boldsymbol{d}, \widetilde{\boldsymbol{F}}). \tag{7.2}$$

In our method, we adopt the recently introduced fully attention-based generalizable NeRF method, GNT [187], as both the backbone and the baseline. GNT shares a similar paradigm with (7.2) but employs transformers [188] to aggregate pixel-level features into $\widetilde{\boldsymbol{F}}$. It uses a *view transformer* to fuse projected pixel-level features from reference views, and a *ray transformer* to combine features from different points along a ray, eliminating the need for volume rendering. Further details about GNT can be found in [187]. We also demonstrate that our approach can be extended to other generalizable NeRF models, as discusses in Section 7.3.2.

7.2.2 Scene-level semantic representation

Both encoder-based generalizable NeRF models [224, 13, 183] and their attention-based counterparts [187, 173] mainly rely on pixel-level feature representations. While effective, this approach restricts their capability for a holistic scene understanding, especially when reference views are scarce. This limitation also exacerbates challenges in resolving depth ambiguities between points along the rays, a problem that becomes more pronounced with fewer reference views.

To address these challenges, we introduce semantic representations aimed at enriching the scene-level understanding. We utilize a shared CNN encoder and apply a Global Average Pooling (GAP) to its

(a) “L: green, R: red" (b) “L: red, R: green"

Figure 7.2: An illustration of conflicting semantic meanings from multiple viewpoints of the same object. When observing the cup from distinct angles, the features retain spatial information but are inconsistent in the scene-level semantic understanding.

C-dimensional output feature map, generating N *global* feature vectors $\{\boldsymbol{S}_n\}_{n=1}^{N}$ corresponding to each input view. These feature vectors are then averaged to form a unified scene-level representation $\boldsymbol{S}$, *i.e*,

$$\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{S}_n \in \mathbb{R}^C. \tag{7.3}$$

In GNT [187], which uses a view transformer to aggregate pxiel-level features into an L-dimensional vector $\widetilde{\boldsymbol{F}}$, we extend this by concatenating $\widetilde{\boldsymbol{F}}$ with $\boldsymbol{S}$ to construct a *global-local* embedding $\boldsymbol{E}$, as formulated by:

$$\boldsymbol{E} = \text{Concat}(\widetilde{\boldsymbol{F}}, \boldsymbol{S}) \in \mathbb{R}^{L+C}. \tag{7.4}$$

This combined embedding $\boldsymbol{E}$ is then subjected to the standard self-attention mechanism in GNT [187]. This approach enables the scene-level semantic representation ($\boldsymbol{S}$) to integrate with per-point features ($\widetilde{\boldsymbol{F}}$), offering a more nuanced understanding at both levels. It also allows each point to selectively draw from the scene-level information. To maintain dimensional consistency across the input and output layers of multiple transformer modules, we employ a two-layer MLP to project the enhanced features back to the original dimension L of the per-point embedding $\widetilde{\boldsymbol{F}}$.

7.2.3 Calibration of semantic representation

The integration of the scene-level semantic representation $\boldsymbol{S}$, generated through simple averaging of global feature vectors as in (7.3), improves rendering quality. However, this approach has limitations when dealing with multiple views. As illustrated in Figure 7.2, viewing the same object from distinct angles may retain spatial attributes but can lead to conflicting semantic meanings. Merely averaging these global feature vectors without accounting for camera positions can result in a distorted scene-level understanding.

To mitigate this inconsistency, we propose a semantic calibration technique using feature rotation. This adjustment aligns the semantic representation across different camera poses. Our inspiration comes from the use of camera pose projection in computing the fused pixel-level feature $\widetilde{\boldsymbol{F}}$ and is further motivated by [143], which demonstrates that explicit rotation operations in feature spaces are feasible. Unlike point clouds in [143] that inherently lack a defined canonical orientation, NeRF explicitly encodes differences between camera viewpoints, thereby enabling precise calibration between the reference and target images.

Building on this observation, we calculate calibrated semantic representations $\{\widetilde{\boldsymbol{S}}_n\}_{n=1}^N$ from the N original semantic representations $\{\boldsymbol{S}_n\}_{n=1}^N$ derived from the reference views. We accomplish this by leveraging their respective rotation matrices $\{\boldsymbol{T}_n\}_{n=1}^N$ to model the rotational variations between each input view and the target view. The alignment of the original semantic features is performed as follows:

$$\widetilde{\boldsymbol{S}}_n = \boldsymbol{\mathcal{P}}(\boldsymbol{T}_n \cdot \boldsymbol{\mathcal{P}}^{-1}(\boldsymbol{S}_n)), \quad \text{where } \boldsymbol{T}_n = \boldsymbol{T}_{\text{out}}^{\text{w2c}} \cdot \boldsymbol{T}_n^{\text{c2w}}. \tag{7.5}$$

Here, $\boldsymbol{T}_n^{\text{c2w}}$ is the inverse of the extrinsic matrix used for $\boldsymbol{I}_n$, and $\boldsymbol{T}_{\text{out}}^{\text{w2c}}$ is the extrinsic matrix for the target view. $\boldsymbol{\mathcal{P}}(\cdot)$ and $\boldsymbol{\mathcal{P}}^{-1}(\cdot)$ are the flattening and inverse flattening operations, which reshape the feature to a 1D vector of shape 1-by-C and a 2D matrix of shape 3-by-$\frac{C}{3}$, respectively.

Note that for the extrinsic matrix, we consider only the top-left 3×3 submatrix that accounts for rotation. Using GAP to condense feature maps of various sizes into a 1-by-C feature vector eliminates the

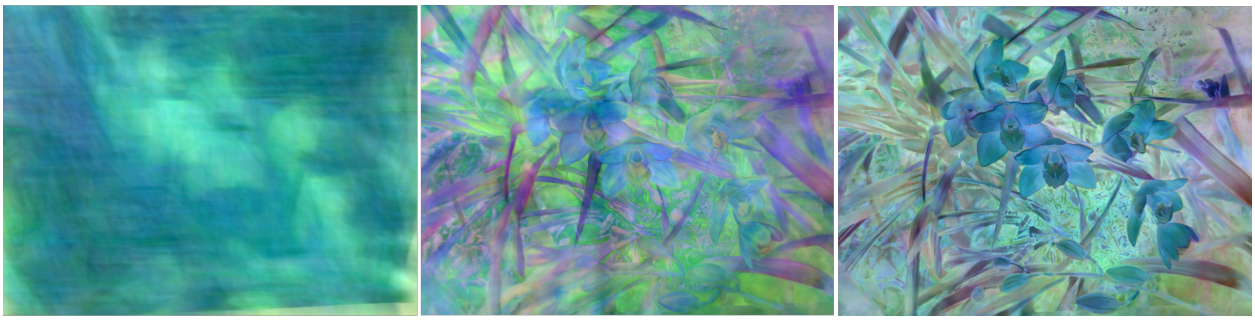


Figure 7.3: Visualization of decoded feature maps for "orchid" in LLFF dataset, produced by *ray transformers* [187] at different stages. From left to right, the transformer stages increase in depth.

need for scaling parameters in the semantic representation. As a result, modeling the intrinsic matrix is unnecessary, assuming no skewing, making our approach adaptable to different camera configurations.

With the calibrated semantic features $\{\widetilde{\boldsymbol{S}}_n\}_{n=1}^N$ for each reference view, we average these, similar to (7.3), to obtain the calibrated scene-level semantic representation $\widetilde{\boldsymbol{S}}$, *i.e*,

$$\widetilde{\boldsymbol{S}} = \frac{1}{N}\sum_{n=1}^{N} \widetilde{\boldsymbol{S}}_n \in \mathbb{R}^C. \tag{7.6}$$

Finally, akin to (7.4), we concatenate the pixel-level fused feature $\widetilde{\boldsymbol{F}}$ with the calibrated scene-level semantic representation $\widetilde{\boldsymbol{S}}$ to form the final global-local embedding $\widetilde{\boldsymbol{E}}$:

$$\widetilde{\boldsymbol{E}} = \text{Concat}(\widetilde{\boldsymbol{F}}, \widetilde{\boldsymbol{S}}) \in \mathbb{R}^{L+C}. \tag{7.7}$$

This unified embedding then feeds into ray transformers, passing through standard self-attention mechanisms. In the original GNT [187], multiple view transformers and ray transformers are stacked alternately for sequential feature processing. The last ray transformer integrates features from multiple points along a ray to yield the final RGB value. We denote the corresponding feature representations at stage k as $\widetilde{\boldsymbol{F}}^{(k)}$ and $\widetilde{\boldsymbol{E}}^{(k)}$. Notably, the calibrated semantic representation $\widetilde{\boldsymbol{S}}$ remains constant across these stages.

7.2.4 Sequential refinement

While leveraging $\widetilde{\boldsymbol{S}}$ improves consistency, a single, uniform $\widetilde{\boldsymbol{S}}$ may not be adequate for deeper layers that demand more nuanced details. In fact, we find that deeper transformers capture finer details compared to shallower ones, as shown in Figure 7.3. To address this limitation, we introduce a sequential semantic feature refinement module that progressively enriches features at each stage. Specifically, we learn the residual $\boldsymbol{\Delta}^{(k)}$ to update $\widetilde{\boldsymbol{S}}$ at each stage k as follows:

$$\widetilde{\boldsymbol{S}}^{(k+1)} \leftarrow \widetilde{\boldsymbol{S}}^{(k)} + \boldsymbol{\Delta}^{(k)}. \tag{7.8}$$

Here, $\boldsymbol{\Delta}^{(k)}$ is calculated by first performing specialized cross-attentions between $\widetilde{\boldsymbol{S}}^{(k)}$ and the original, uncalibrated per-frame semantic features $\{\boldsymbol{S}_n\}_{n=1}^{N}$ (see Figure 5.1), followed by their summation. Our goal is to fuse information from different source views to enrich the scene-level semantic representation with features from each reference frame. With this sequential refinement, we combine $\widetilde{\boldsymbol{S}}^{(k)}$ with $\widetilde{\boldsymbol{F}}^{(k)}$ at each stage, yielding a stage-specific global-local embedding $\widetilde{\boldsymbol{E}}^{(k)}$, which completes our approach.

Discussion. In scenarios with few reference views, especially when limited to just one, the primary issue is inaccurate depth estimation, resulting in depth ambiguity [29]. This compromises the quality of images when rendered from novel viewpoints. Despite this, essential visual information generally remains accurate across different camera poses. Incorporating our proposed scene-level representation improves the understanding of the overall scene layout [18], distinguishing our approach from existing generalizable NeRF models that predict pixels individually. The advantage of our approach is its holistic view; the semantic representation enriches per-pixel predictions by providing broader context. This semantic constraint ensures that fewer abrupt changes between adjacent points. Consequently, it leads to more reliable depth estimations, making the images rendered from limited reference views more plausible.

7.2.5 Training objectives

During training, we employ three different loss functions:

MSE loss. The Mean Square Error (MSE) loss is the standard photometric loss used in NeRF [128]. It computes the MSE between the actual and predicted pixel values.

Central loss. To ensure frame-wise calibrated semantic features $\{\widetilde{\boldsymbol{S}}_n\}_{n=1}^{N}$ are consistent when projected onto the same target view, we introduce a central loss, defined as:

$$\mathcal{L}_{\text{central}} = \frac{1}{N}\sum_{n=1}^{N}\left\|\widetilde{\boldsymbol{S}}_n - \widetilde{\boldsymbol{S}}\right\|_1. \tag{7.9}$$

Point-wise perceptual loss. During the rendering of a bath of pixels in a target view, we inpaint the ground-truth image by replacing the corresponding pixels with the predicted ones. Then, a perceptual loss [86] is computed between the inpainted image and the target image to guide the training process at the whole-image level.

The final loss function is formulated as follows:

$$\mathcal{L} = \mathcal{L}_{\text{MSE}} + \lambda_1\mathcal{L}_{\text{central}} + \lambda_2\mathcal{L}_{\text{perc}}. \tag{7.10}$$

Empirically, we set $\lambda_1 = 1$ and $\lambda_2 = 0.001$, following [107].

7.3 Experiments

7.3.1 Experimental setups

For the experimental setups, we begin by describing the datasets used in our experiments. This is followed by implementation details of our proposed method and the baseline methods we employed for comparison.

Datasets. Firstly, following [187], we construct our training data from both synthetic and real data. This collection includes scanned models from Google Scanned Objects [34], RealEstate10K [247], and hand-held phone captures [192]. For evaluation, we utilize real data encompassing complex scenes from sources such as LLFF [127], Shiny [208], and mip-NeRF 360 [4]. Additionally, we train and test our model using the recently released MVImgNet dataset [227]. We adhere to the official split, focusing on examples from the *containers* category, and select 2,500 scenes for training. During inference, we choose 100 scenes, using their first images as target views and the spatially nearest images as references. Since MVImgNet does not provide camera poses, we utilize COLMAP [156, 157] to deduce the camera positions within these scenes.

Implementation details. CaesarNeRF is built upon GNT [187], for which we maintain the same configuration, setting the *ray* and *view* transformers stack number (K) to 8 for generalizable setting and 4 for single-scene setting. The feature encoder extracts bottleneck features, applies GAP, and then uses a fully connected (FC) layer to reduce the input dimension C to 96. Training involves 500,000 iterations using the Adam optimizer [91], with learning rate set at 0.001 for the feature encoder and 0.0005 for CaesarNeRF, halving them every 100,000 iterations. Each iteration samples 4,096 rays from a single scene. In line with [187], we randomly choose between 8 to 10 reference views for training, and 3 to 7 views when using the MVImgNet [227].

Baseline methods. We compare CaesarNeRF with several state-of-the-art methods suited for generalizable NeRF applications, including earlier works such as MVSNeRF [13], PixelNeRF [224], and IBRNet [192], alongside more recent ones, including GPNR [173], NeuRay [117], GNT [187], GeoNeRF [85] and MatchNeRF [20].

7.3.2 Results and analysis

We compare results in two settings: a generalizable setting, where the model is trained on multiple scenes without fine-tuning during inference for both few and all reference view cases, and a single-scene setting

Table 7.1: Results for generalizable scene rendering on LLFF with few reference views. GeoNeRF, MatchNeRF, and MVSNeRF necessitate variance as input, defaulting to 0 for single-image cases, hence their results are not included for 1-view scenarios.

Input	Method	PSNR (↑)	LPIPS (↓)	SSIM (↑)
1-view	PixelNeRF [224]	9.32	0.898	0.264
	GPNR [173]	15.91	0.527	0.400
	NeuRay [117]	16.18	0.584	0.393
	IBRNet [192]	16.85	0.542	0.507
	GNT [187]	16.57	0.500	0.424
	Ours	18.31	0.435	0.521
2-view	PixelNeRF [224]	11.23	0.766	0.282
	GPNR [173]	18.79	0.380	0.575
	NeuRay [117]	17.71	0.336	0.646
	GeoNeRF [85]	18.76	0.473	0.500
	MatchNeRF [20]	21.08	0.272	0.689
	MVSNeRF [13]	19.15	0.336	0.704
	IBRNet [192]	21.25	0.333	0.685
	GNT [187]	20.88	0.251	0.691
	Ours	21.94	0.224	0.736
3-view	PixelNeRF [224]	11.24	0.671	0.486
	GPNR [173]	21.57	0.288	0.695
	NeuRay [117]	18.26	0.310	0.672
	GeoNeRF [85]	23.40	0.246	0.766
	MatchNeRF [20]	22.30	0.234	0.731
	MVSNeRF [13]	19.84	0.314	0.729
	IBRNet [192]	23.00	0.262	0.752
	GNT [187]	23.21	0.178	0.782
	Ours	23.45	0.176	0.794

where the model is trained and evaluated on just one scene. Following these comparisons, we conduct ablation studies and test the generalizability of our method with other state-of-the-art approaches.

Generalizable rendering. In the generalizable setting, we adopt two training strategies. First, we train the model on multiple datasets as described in Section 7.3.1 and evaluate on LLFF [127], Shiny [208] and mip-NeRF 360 [4] datasets. In addition, the model is trained and tested on the MVImgNet [227] for object-centric generalizability.

Table 7.2: Results for generalizable scene rendering on Shiny with few reference views.

Input	Method	PSNR (↑)	LPIPS (↓)	SSIM (↑)
1-view	IBRNet [192]	14.93	0.625	0.401
	GNT [187]	15.99	0.548	0.400
	Ours	17.57	0.467	0.472
2-view	MatchNeRF [20]	20.28	0.278	0.636
	MVSNeRF [13]	17.25	0.416	0.577
	IBRNet [192]	18.40	0.400	0.595
	GNT [187]	20.42	0.327	0.617
	Ours	21.47	0.293	0.652
3-view	MatchNeRF [20]	20.77	0.249	0.672
	MVSNeRF [13]	18.55	0.343	0.645
	IBRNet [192]	21.96	0.281	0.710
	GNT [187]	22.47	0.247	0.720
	Ours	22.74	0.241	0.723

(a) LLFF, Shiny, and mip-NeRF 360. The results for few-reference view scenarios on these datasets are shown in Tables 7.1, 7.2 and 7.3, respectively. Methods like MatchNeRF [20], MVSNeRF [222], and GeoNeRF [85] require at least two reference views. On the LLFF dataset, all methods experience a performance decline as the number of views decreases. CaesarNeRF, however, consistently outperforms others across varying reference view numbers, with the performance gap becoming more significant with fewer views. For example, with 3 views, while IBRNet [192] and GNT [187] have comparable PSNRs, CaesarNeRF demonstrates a more substantial lead in LPIPS and SSIM metrics.

Similar patterns are observed on the Shiny [208] and mip-NeRF 360 [4] datasets. We apply the highest-performing methods from the LLFF evaluations and report the results for those that produce satisfactory outcomes with few reference views. CaesarNeRF maintains superior performance throughout. Notably, for complex datasets like mip-NeRF 360 [4], which have sparse camera inputs, the quality of rendered images generally decreases with fewer available reference views. Nonetheless, CaesarNeRF shows the most robust performance compared to the other methods.

(b) MVImgNet. We extend our comparison of CaesarNeRF with GNT [187] and IBRNet [192] on the MVImgNet dataset, focusing on object-centric scenes, as shown in Table 7.4. We examine a variant of

Table 7.3: Results for generalizable scene rendering on mip-NeRF 360 with few reference views.

Input	Method	PSNR (↑)	LPIPS (↓)	SSIM (↑)
1-view	IBRNet [192]	14.12	0.682	0.283
	GNT [187]	13.48	0.630	0.314
	Ours	15.20	0.592	0.350
2-view	MatchNeRF [20]	17.00	0.566	0.392
	MVSNeRF [13]	14.23	0.681	0.366
	IBRNet [192]	16.24	0.618	0.360
	GNT [187]	15.21	0.559	0.370
	Ours	17.05	0.538	0.403
3-view	MatchNeRF [20]	17.26	0.551	0.407
	MVSNeRF [13]	14.29	0.674	0.406
	IBRNet [192]	17.70	0.555	0.420
	GNT [187]	15.59	0.538	0.395
	Ours	17.55	0.512	0.430

CaesarNeRF where semantic calibration is substituted with simple feature averaging from multiple frames. While the performance of all methods improves with more views, CaesarNeRF consistently outperforms GNT and IBRNet. Notably, CaesarNeRF with feature averaging surpasses GNT in 1-view case but lags with additional views, implying that the absence of calibration lead to ambiguities when rendering from multiple views.

Per-scene optimization. Beyond the multi-scene generalizable setting, we demonstrate per-scene optimization results in Table 7.5. Testing across 8 categories from the LLFF dataset [127], we calculate the average performance over these scenes. CaesarNeRF consistently outperforms nearly all state-of-the-art methods in the comparison, across all three metrics, showing a significant improvement over our baseline method, GNT [187].

Adaptability. To test the adaptability of our Caesar pipeline, we apply it to two other state-of-the-art methods that use *view transformers*, namely MatchNeRF [20] and IBRNet [192]. We demonstrate in Table 7.6 that our enhancements in scene-level semantic understanding significantly boost the performance of these methods across all metrics. This indicates that the Caesar framework is not only beneficial in our CaesarNeRF, which is based on GNT [187], but can also be a versatile addition to other NeRF pipelines.

Table 7.4: Results on MVImgNet across varying numbers of reference views. 'C.' represents the use of calibration before averaging.

		PSNR (↑)	LPIPS (↓)	SSIM (↑)
1-view	IBRNet	19.14	0.458	0.595
	GNT	22.22	0.433	0.678
	Ours *w/o* C.	23.61	0.371	0.718
	Ours	24.28	0.334	0.747
2-view	IBRNet	24.38	0.266	0.818
	GNT	26.94	0.236	0.850
	Ours *w/o* C.	26.34	0.274	0.817
	Ours	27.34	0.215	0.856
3-view	IBRNet	25.53	0.203	0.858
	GNT	27.41	0.206	0.870
	Ours *w/o* C.	27.10	0.228	0.850
	Ours	27.82	0.190	0.875
4-view	IBRNet	25.99	0.190	0.867
	GNT	27.51	0.197	0.875
	Ours *w/o* C.	27.30	0.210	0.862
	Ours	27.92	0.181	0.881
5-view	IBRNet	26.12	0.188	0.867
	GNT	27.51	0.194	0.876
	Ours *w/o* C.	27.34	0.203	0.865
	Ours	27.92	0.179	0.882

Ablation analysis. We conduct ablation studies on the "orchid" scene from the LLFF dataset, with findings detailed in Table 7.7. Testing variations in representation and the impact of the sequential refinement and calibration modules, we find that increasing the latent size in GNT yields marginal benefits. However, incorporating even a modest semantic representation size distinctly improves results. The length of the semantic representation has a minimal impact on quality. Our ablation studies indicate that while sequential refinement and calibration each offer slight performance gains, their combined effect is most significant. In a single-scene context, semantic information is effectively embedded within the representation, making the benefits of individual modules subtler. Together, however, they provide a framework where sequential refinement can leverage calibrated features for deeper insights.

Table 7.5: Results of per-scene optimization on LLFF, in comparison with state-of-the-art methods.

Method	PSNR (↑)	LPIPS (↓)	SSIM (↑)
LLFF [127]	23.27	0.212	0.798
NeRF [129]	26.50	0.250	0.811
NeX [208]	27.26	0.179	0.904
GNT [187]	27.24	0.087	0.889
Ours	27.64	0.081	0.904

Table 7.6: Results on LLFF for few-shot generalization after adapting Caesar to other baseline methods.

Method	Input	PSNR (↑)	SSIM (↑)	LPIPS (↓)
MatchNeRF [20]	2-view	20.59	0.775	0.276
	3-view	22.43	0.805	0.244
Caesar-MatchNeRF	2-view	21.55	0.782	0.268
	3-view	22.98	0.824	0.242
IBRNet [192]	1-view	16.85	0.507	0.542
	2-view	21.25	0.685	0.333
	3-view	23.00	0.752	0.262
Caesar-IBRNet	1-view	17.76	0.543	0.500
	2-view	22.39	0.740	0.275
	3-view	23.67	0.772	0.242

Visualizations. We present our visualization results in Figure 7.4, where we compare our method with others using one or two views from the LLFF dataset. These visualizations highlight that in scenarios with few views, our method significantly surpasses the competitors, particularly excelling when only a single view is available. In such cases, CaesarNeRF demonstrates enhanced clarity, with sharper boundaries and more distinct objects.

Depth estimation. In addition to RGB image predictions, we also explore depth prediction on the LLFF [127] dataset, specifically in scenarios with few reference views, such as one or two. We compare the performance of CaesarNeRF and GNT [187] in Table 7.5. We find that GNT struggles to accurately represent the relative positions of objects within the scene when provided with few reference views. For instance, in the flower case with just one view, CaesarNeRF accurately shows that the flower is closer to

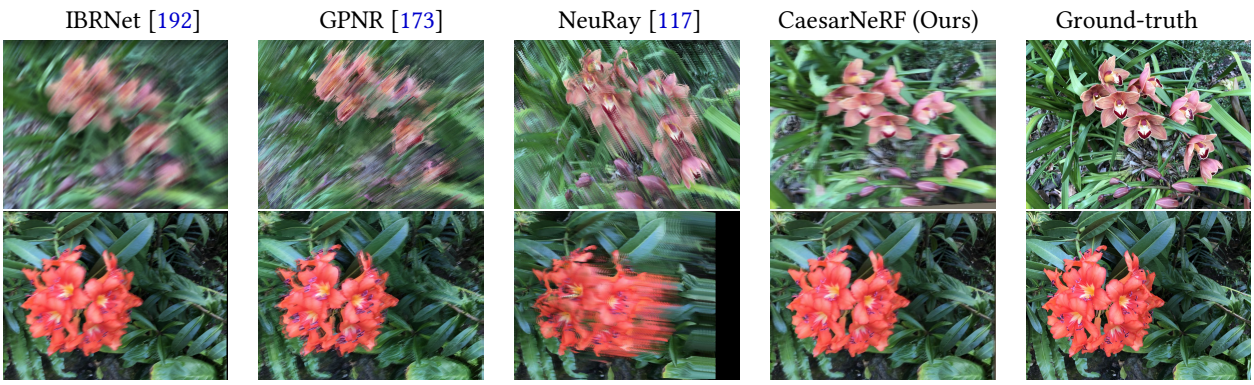
Table 7.7: Ablations on the semantic representation length $\boldsymbol{R}$, sequential refinement (Seq.) and calibration (Cali.). 'Ext.' denotes the extension of per-pixel representation to a length of 64 in GNT.

Model Variations			PSNR (↑)	LPIPS (↓)	SSIM (↑)
$\boldsymbol{R}$ len.	Seq.	Cali.			
(Baseline GNT)			20.93	0.185	0.731
Ext.			20.85	0.173	0.735
+32			21.43	0.152	0.763
+64			21.49	0.149	0.766
+96			21.46	0.150	0.766
+128			21.49	0.147	0.763
+ 96	✓		21.53	0.146	0.770
+ 96		✓	21.51	0.147	0.769
+ 96	✓	✓	21.67	0.139	0.781

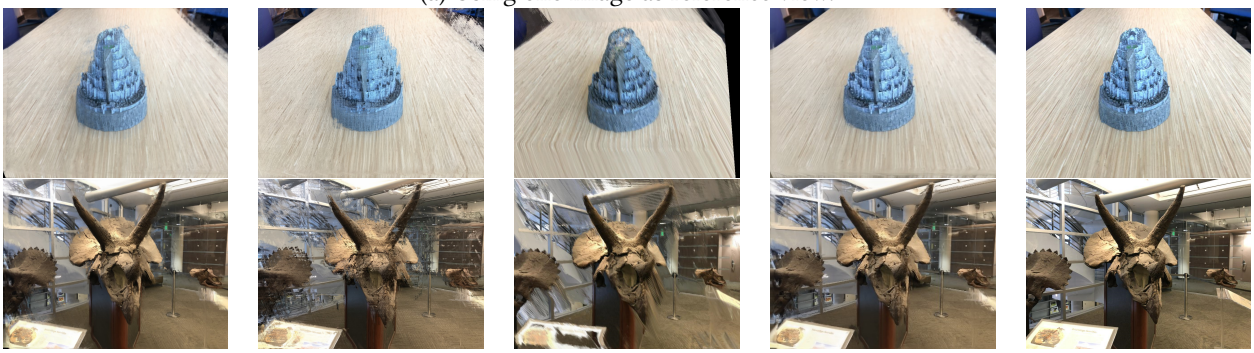
the camera than the leaves in the background, whereas GNT doesn't show a distinguishable difference in depth prediction.

Furthermore, the depth estimations provided by CaesarNeRF are more consistent. In the horn example involving two views, CaesarNeRF offers better boundary delineation, particularly when confronted with reflective surfaces like the glass in the background. While CaesarNeRF performs well in capturing the relative depths within the scene, it tends to predict the background as being farther from the camera. The absolute depth values are not accurate enough to resolve all ambiguities in the reconstructed depth, constituting a potential limitation that could be addressed in future work.

Semantic analysis for $\tilde{\boldsymbol{S}}$. As the averaged calibrated semantic representation $\tilde{\boldsymbol{S}}$ includes information from frames across the same scene, we conduct further analysis to determine if such a feature from the encoder indeed encompasses semantic details about the scene. We use examples from the LLFF test set [127] to match with LLFF training examples and examine the highest and lowest response based on their L-2 distance between features from two scenes. Since we lack specific labels, the L-2 distance serves as an implicit reflection of the similarity between two vectors. A smaller distance indicates more similarity between them, and vice versa. To extract the scene-level representation for each scene, we consider the



(a) Using one image as reference view.



(b) Using two images as reference view.

Figure 7.4: Comparative visualization of our proposed method against other state-of-the-art methods.

first image of each category in the LLFF training and test sets as a reference image and apply CaesarNeRF to extract the scene-level representation for the nearest ten views surrounding the reference image.

We present two examples, "room" and "fortress", accompanied by the first image of the category with the top-2 highest and lowest responses from the LLFF training set in Figure 7.6. In extracting the scene-level representation, the representation predominantly emphasizes structural information as well as certain similar object categories. For instance, with the source image "room", the highest responses align with table-like structures, and the scene is mainly object-centric. Conversely, the lowest responses display images of flowers or a pond in open spaces, categories not present in the source images. A similar observation applies to the second example, "fortress", where the top two responses reflect spiral structures like playground slides and bicycle pegs. In contrast, the lowest two responses from the LLFF training scenes originate from scenes with displayed objects. It's worth noting that color is not the paramount factor for

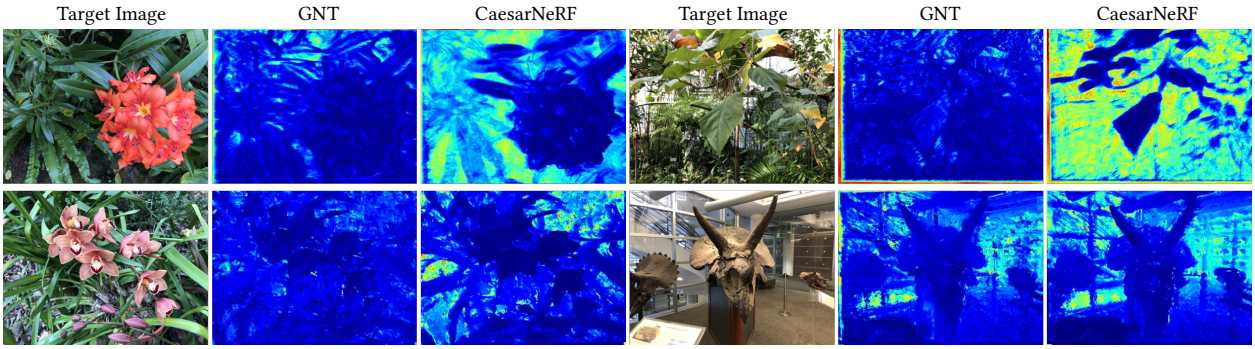


Figure 7.5: Depth estimation prediction using one reference view (first row) and two reference views (second row) as input from LLFF comparing CaesarNeRF with GNT.

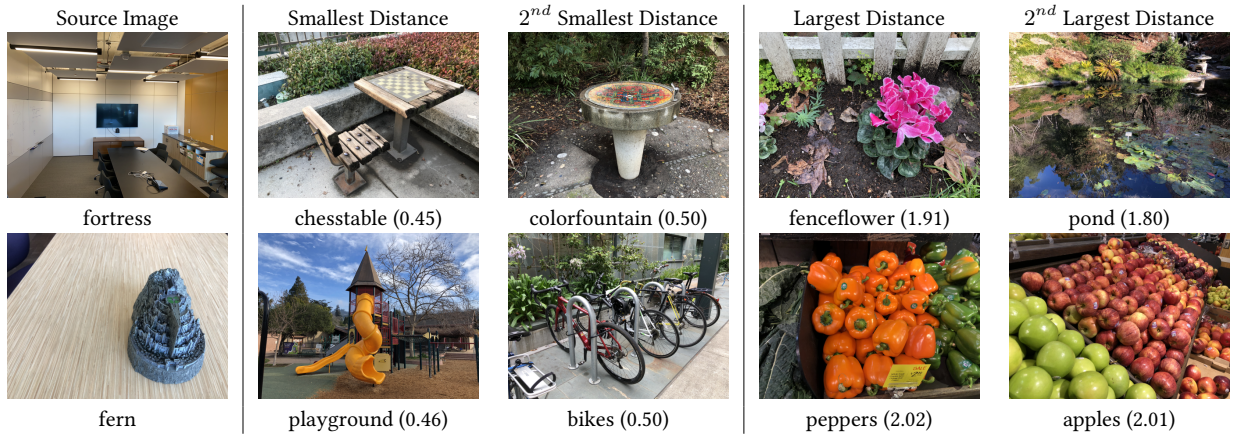


Figure 7.6: Largest and smallest distances for two examples from LLFF test split when matching with training scenes. Numbers ($\times 10^{-2}$) in the brackets are the L-2 distance to the source image.

scene-level semantic representation. This is evident from the large yellow and dark patches in the last example (pond) of the first row, which still has the second-lowest response when matched with the source image.

7.4 Conclusion and limitation

In this chapter, we introduce CaesarNeRF, a few-shot and generalizable NeRF pipeline that combines scene-level semantic with per-pixel feature representations, aiding in rendering from novel camera positions with limited reference views. We calibrate the semantic representations across different input views and employ a sequential refinement network to offer distinct semantic representations at various levels. Our method

has been extensively evaluated on a broad range of datasets, exhibiting state-of-the-art performance in both generalizable and single-scene settings.

Limitations. CaesarNeRF could be further improved by integrating explicit depth information and generative capabilities, which could provide a richer basis for rendering from novel views with few reference images.

Chapter 8

SEAS: Shape-assisted Re-Identification with Implicit Representation

8.1 Introduction

With per-pixel level representations capturing better body shapes, we include the information to augment the task of person re-identification with raw frames as input. Person re-identification [93, 202], which aims to identify an individual from a collection of pedestrian images or videos captured by non-overlapping cameras, is a crucial task for biometric understanding. Existing methods [65, 90, 260, 200] primarily focus on a person's appearance, which can be affected by environmental variations as the appearance is often intertwined with the background. We explore the use of 3-D body shape as supervision to enhance the human-centric appearance and demonstrate significant improvement on public datasets compared to other state-of-the-art methods, as illustrated in Figure 8.1.

When introducing a second modality, existing re-identification methods [111, 15, 233, 146] often employ a separate branch alongside appearance for person identification. While these methods can enhance the features for re-identification, they encode each modality independently, thereby diminishing the integration of the two input modalities and weakening their connection. Additionally, the extra encoder required introduces new parameters to the network, leading to increased model size and computational cost.

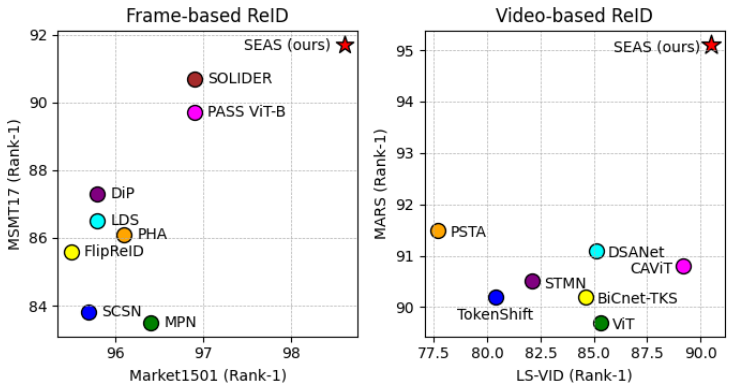


Figure 8.1: Rank-1 accuracy of using SEAS on ResNet-50 backbone on frame-based (MSMT17 [203] and Market1501 [239]). and PSTA as encoders compared with other state-of-the-art methods on and video-based (LS-VID [95] and MARS [238]) person re-identification datasets.

We introduce the use of human body **S**hap**E**-**A**ligned **S**upervision, abbreviated as SEAS, to enhance appearance-based re-identification methods. Instead of using a secondary modality as model input, we utilize it to guide the generation of identity features with a trainable body shape extractor. This extractor takes identity feature maps from the identity encoder and converts them into pixel-level features that represent 3-D body shapes. Throughout training, we direct the encoder to augment feature extraction with pixel-level shape-related information by supervising the generation of body shapes with an external pretrained model. This also causes the identity feature map to maintain more appearance information across a wider area in the body image, as it is the input for decoding pixel-level body shape features. During inference, the extractor is discarded, allowing the encoder to adeptly extract both shape and appearance-related information without incurring extra time cost. This supervision can be adapted to various encoders, as we demonstrate later in Section 8.3.3.

As the model input can either be a single frame or a video consisting of multiple consecutive frames, we employ different strategies to leverage the 3-D body shape in augmenting appearance. For frame-based person re-identification, where the input is a single RGB image, we incorporate a pixel-level implicit representation as supplementary guidance to provide body shape details in conjunction with appearance. During training, after the encoder produces the feature map, we attach a series of deconvolutional layers,

functioning as the extractor, to upscale this map to a pixel-level representation. We supervise the decoded features from the extractor using the feature map generated by the PIFu [152] encoder with pixel-level shape guidance.

Different from a single frame, video input provides a consistent appearance across multiple frames since they depict the same person. Therefore, we integrate cross-frame appearance consistency with body shape alignment during training based on the per-pixel implicit body representation. We project the pixel-level features of each point on the body in every frame of the same video onto a unified body model and reduce the variance of the features at the same point, enforcing point-level consistency. This alignment generates a shared body model that captures appearance in 3-D space, allowing the identity encoder to extract features for both appearance and body shape with temporal consistency.

As SEAS can be applied to both frame and video-based re-identification, we assess it with both settings. For frame-based re-identification, we evaluate SEAS with ResNet-50 on Market1501 [239] and MSMT17 [203]. For video-based re-identification, we test it with PSTA [200] on MARS [238] and LS-VID [95]. Our results show a 32.8% and 27.2% relative reduction in rank-1 errors for frame and video-based re-identification compared to state-of-the-art methods.

In summary, our contributions are as follows: 1) we introduce SEAS, applying shape-aligned supervision to person re-identification, utilizing 3-D body features in addition to appearance without additional cost during inference; 2) we propose a pixel-level feature alignment across frames in video-based re-identification for temporal consistency; and 3) we present the superior performance and adaptability to other encoders of using SEAS for both image and video-based re-identification through extensive experiments.

8.2 Method

As an enhancement to existing appearance-based person re-identification models, SEAS decodes body shape with an extractor that follows the identity feature encoder generating convolutional features for re-identification, as illustrated in Figure 8.2. During inference, only the identity encoder is retained, ensuring there is no additional computational cost.

In the following subsections, we start by providing a brief overview of the person re-identification task and the feature encoders used to generate identity features in Section 8.2.1. We then detail SEAS in Section 8.2.2 and provide an in-depth discussion on the training and inference phases with further analysis in Section 8.2.3.

8.2.1 Identity Feature Encoder

When taking a video $\boldsymbol{V} = \{\boldsymbol{I}_i\}$ or a single frame image $\boldsymbol{I}$ as input, person re-identification aims to encode the corresponding identity feature $\boldsymbol{F}^{ID}$ from the input modality and use it to search in the gallery for the best match. The label of the feature with the highest response in the gallery is considered as the prediction. As we have two different types of input, videos and single frames, we have two model variations for the feature encoding.

To process single-frame inputs, we first resize the input image to 256×128 and employ a ResNet-50 [67] initialized with parameters trained on ImageNet [151] to encode the image $\boldsymbol{I}$ into its corresponding feature map $\boldsymbol{F}^{map}$. This map is the output from the last convolutional layer in the encoder, which retains spatial information. We then apply a global average pooling (GAP) layer to pool the $\boldsymbol{F}^{map}$, transforming the $H \times W \times C$ feature maps into a $1 \times 1 \times C$ feature vector $\boldsymbol{F}^{ID}$ used for re-identification following

$$\boldsymbol{F}^{ID} = \text{GAP}(\boldsymbol{F}^{map}). \tag{8.1}$$

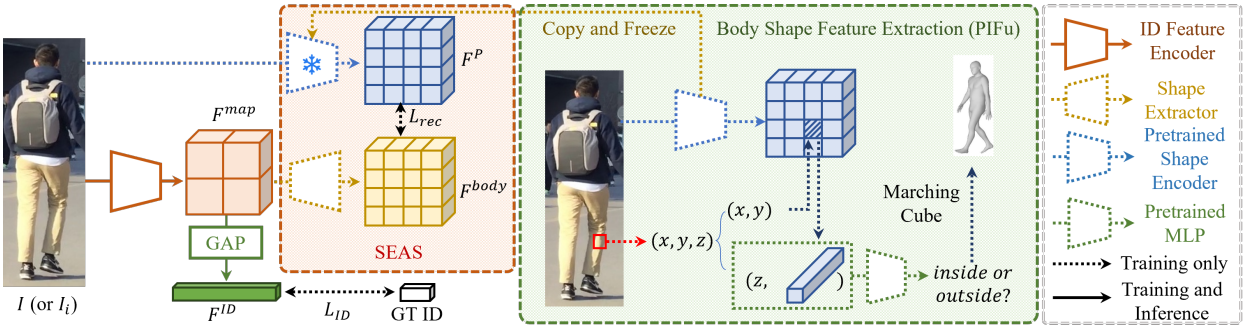


Figure 8.2: Person re-identification uses implicit body shape extraction as supervision, with (left) the identity feature encoder along with the SEAS pipeline, and (right) a brief overview of the PIFu [152] pipeline. Trapezoids in the figures are trainable models and GAP stands for Global Average Pooling. Dotted lines are used only during training and are excluded during inference.

Similar to the extraction of the feature map from a single image, when using a video as input with sequential images $\{\boldsymbol{I}_i\}$, we utilize a shared ResNet-50 [67] to encode each frame into its corresponding feature maps $\{\boldsymbol{F}_i^{map}\}$. We follow PSTA [200] and construct the pyramid spatial and temporal attention to aggregate features from different frames, resulting in the final feature map $\boldsymbol{F}^{map}$. As the encoder is not the contribution of this work, more details are available in [200]. We then follow Equation 8.1 to transform the feature map into a feature vector for identification.

8.2.2 SEAS: Shape-Aligned Supervision

To use the 3-D shape information as guidance to input frames for re-identification, we introduce SEAS, shape-aligned supervision, to enhance the identity encoder with body shape alongside appearance. We apply SEAS for frame-based and video-based tasks separately as single frames primarily emphasize extracting features from an individual image, while frames within a video are intrinsically interlinked and can be represented by a shared 3-D body model. Given these considerations, we employ pixel-level implicit representation [152, 153] to supervise single frame input. For videos, we introduce per-pixel feature calibration atop implicit features to emphasize temporal consistency.

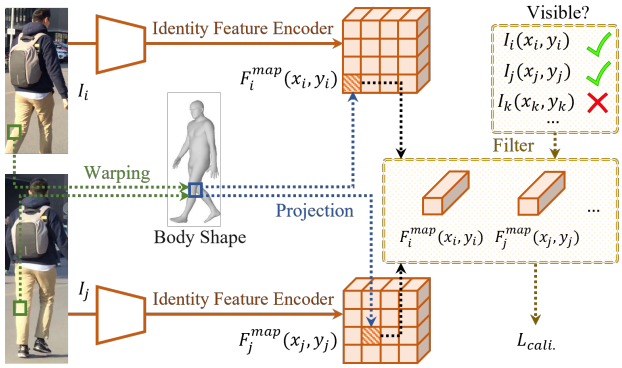


Figure 8.3: Feature calibration across different frames. We use a point near the left knee as an example. We warp the 2-D points using a shared 3-D body shape and project them onto the feature maps F^{map} to extract point-level features with interpolation, followed by a calibration loss $\mathcal{L}_{cali.}$ to reduce the variance of features for corresponding points across different frames that are visible and mapped to the same location on the shared body shape.

SEAS for Frame-based Re-Identification. We integrate pixel-level representations extracted from pretrained PIFu [152] as supervision for framewise body shape generated from SEAS. PIFu [152], as depicted on the right of Figure 8.2, characterizes body shape using a feature map that implicitly records whether points in 3-D space are inside or outside the object. Given an input image $\boldsymbol{I}$, PIFu extracts per-pixel level features $\boldsymbol{F}^P$ with an encoder. For each pixel, PIFu combines a depth value, denoted as z, with its associated feature $\boldsymbol{F}^P(x, y)$ sampled from the feature map, and uses their concatenation as input to an MLP network to decode the signed distance value at that specific depth. The signed distance [25, 136] indicates the distance to the nearest surface, with the sign determining whether the point is inside or outside the object. By aggregating these dense signed distance values in 3-D space, PIFu can reconstruct the object's surface by locating the zero-value surface.

We employ the extracted pixel-level body shape representation $\boldsymbol{F}^P$ to guide the extraction of body shapes using the encoded identity feature map $\boldsymbol{F}^{map}$. Since $\boldsymbol{F}^{map}$ emanates from the last convolutional layer of the identity feature encoder, it retains the highest level of semantic representation, along with spatial information before global average pooling. Furthermore, as we employ the feature post-pooling directly for identification in frame-based input per Equation 8.1, leveraging $\boldsymbol{F}^{map}$ to decode body shape

information ensures maximal preservation of body features in the final identification representation $\boldsymbol{F}^{ID}$. Given the pooling operations in the identity and PIFu feature encoders, we upscale $\boldsymbol{F}^{map}$ to align with the size of $\boldsymbol{F}^{P}$ as follows:

$$\boldsymbol{F}^{body} = \text{UpConv} \cdot \ldots \cdot \text{UpConv}(\boldsymbol{F}^{map}). \tag{8.2}$$

Here, $\boldsymbol{F}^{body}$ represents the decoded per-pixel feature map for the body shape. The `UpConv` sequence consists of 2-D transposed convolutions, BatchNorm, and ReLU, followed by 2-D convolutions, BatchNorm, and ReLU, in that order. We exclude BatchNorm and ReLU from the final `UpConv` step because $\boldsymbol{F}^{P}$ may include non-positive values. The width and height of the feature map are doubled with each `UpConv` layer, with the total number of layers determined by the size difference between $\boldsymbol{F}^{map}$ and $\boldsymbol{F}^{body}$.

With the extracted body shape feature $\boldsymbol{F}^{body}$ and the corresponding PIFu-encoded feature $\boldsymbol{F}^{P}$, we apply $\mathcal{L}_{rec}$ to train the identity encoder and the extractor in SEAS using the smooth L1 loss following

$$\mathcal{L}_{rec} = \frac{1}{HW} \sum \text{SmoothL1}(\boldsymbol{F}^{body}, \boldsymbol{F}^{P}) \tag{8.3}$$

to compute the average of per-pixel differences across feature maps of size $H \times W$. The smooth L1 loss generates reasonable gradients near the zero point of the differences and prevents the model from overly penalizing values with large deviations from zero.

SEAS for Video-based Re-Identification. Video input includes consistency across frames for the same individual, while an encoder without such correspondence awareness might generate disparate features due to varying poses or image resolutions. Inspired by [140], which suggests that the per-pixel appearance of the same body part can be decoded from the same point-level features even with different poses, maintaining pixel-level feature consistency enforces appearance consistency across frames. Therefore, in addition to SEAS for single frames, we integrate feature calibration for points across multiple frames

that correspond to the same body part, as depicted in Figure 8.3, which ensures temporal consistency at the pixel level.

To align the body shape across different frames, we adopt three steps for feature calibration: 1) extract the per-pixel level features, 2) determine the correspondence between points across various frames and project the per-pixel features onto a shared body shape, and 3) align the features from different frames that represent the same point. We utilize the feature maps from the identity encoder $\{\boldsymbol{F}_i^{map}\}$ as our input since they also maintain the highest level of semantic representations and can avoid the potential conflict between the alignment and $\{\boldsymbol{F}_i^{P}\}$ during supervision.

With the extracted framewise per-pixel features, we establish dense correspondence using SMPL [119], which provides a predefined body shape with 6,890 vertices on the body surface, ensuring each point has a specific order and representation. For different frames in the video $\{\boldsymbol{I}_i\}$, we first warp the images using a shared body shape model to build pixel-level point correspondence between the frames and the shared body shape. We then sample k points on the body shape and locate their corresponding points in each frame, gathering the corresponding pixel-level features using bilinear interpolation from the nearest four points on the feature map. As points may be occluded and therefore invisible, we use the normal of each point to determine visibility in the images and accordingly filter the features.

After collecting the features of these k points from the frames in which they are visible, we calculate the variance of the features for each point and aggregate them following

$$\mathcal{L}_{cali.} = \frac{1}{kn} \sum_{k} \sum_{n} \text{Variance}(\boldsymbol{F}_i^{map}(x_i, y_i)) \tag{8.4}$$

where n represents the number of frames in which the sampled points are visible. By minimizing the variance for the sampled k points, we can integrate the temporal consistency across different frames within a single video in addition to the implicit shape representations $\{\boldsymbol{F}_i^{body}\}$.

8.2.3 Training and Inference

During training, we employ three different losses: the re-identification loss $\mathcal{L}_{ID}$, the reconstruction loss $\mathcal{L}_{rec}$, which supervises the 3-D shape features produced by the SEAS extractor, and the feature calibration loss $\mathcal{L}_{cali.}$ for video-based re-identification, aligning features of the same body part across various frames. We provide more details about $\mathcal{L}_{ID}$ in Section 8.3.1. The overall loss is defined as

$$\mathcal{L} = \mathcal{L}_{ID} + \lambda_1 \mathcal{L}_{rec} + \lambda_2 \mathcal{L}_{cali.} \tag{8.5}$$

where λ_1 and λ_2 are hyperparameters to balance the different loss terms. λ_2 is set to 0 when SEAS is applied exclusively to frame-based re-identification tasks.

During inference, we detach SEAS from the identity encoder, keeping only the encoder for identity feature extraction. We process its output feature F^{map} and convert it into F^{ID} for person identification using global average pooling. After computing the cosine similarity between the gallery and query features, we assign the label of the example with the highest response as our prediction of the query video.

Discussion. As the input to the body shape extractor is the output from the identity feature encoder, the only data used to decode body shapes originates from this encoder. Since pixel-level body shapes are predicted using the information from pixel-level features, this process facilitates a pixel-level integration of appearance and body shape information. This ensures that the feature map $\boldsymbol{F}^{map}$, which is used for re-identification after global average pooling, not only is explicitly shape-aware but also maximally retains the pixel-level appearance information for the body area to decode pixel-level body shapes, instead of focusing on a small patch of it for re-identification. Additionally, when processing video inputs, employing a shared body shape projector aligns different points from various input perspectives into a unified space.

This alignment provides explicit temporal consistency across frames, allowing the encoder to extract temporally coherent information inherently. We verify this with multiple model variations as well as attention visualization, which we present later in Section 8.3.3.

Compared with existing methods [111, 15] that explicitly decode body shapes from images and use them for identification, SEAS differs in two main aspects: 1) SEAS does not explicitly require 3-D reconstruction as guidance for identification, while it preserves it as implicit supervision for pixel-level coherence with appearance. 2) SEAS directly connects the appearance and shape across different frames within a video. Existing methods [111, 15] reconstruct the frame-wise body shapes and do not aggregate the temporal information across different frames, while SEAS can explicitly provide such guidance with variance computation on the same vertex of the SMPL body to offer more appearance consistency without extra computational cost during inference, as only the encoder is needed.

8.3 Experiments

In this section, we first discuss the details and settings of our experiments, including the implementation details, datasets used, and baseline methods in Section 8.3.1, followed by results, analysis, and ablation studies in Section 8.3.2.

8.3.1 Experimental Details

Datasets. For our experiments, we utilize Market1501 [239] with MSMT17 [203] for frame-based person re-identification and MARS [238] in conjunction with LS-VID [95] for video-based person re-identification.

Market1501 [239] and MSMT17 [203] serve as two datasets for the person re-identification public benchmark with single-frame images as input. Market1501 [239] includes 1,501 identities, with 750 designated for training and the remaining 751 for testing, captured across six cameras. The training set, gallery, and query consist of 12,936, 3,368, and 19,732 cropped images, respectively. MSMT17 [203] comprises 4,101

identities, with 1,041 for training and 3,060 for testing. Images are collected from 15 cameras (12 for outdoor scenes and 3 for indoor scenes) with the numbers of images for training, gallery, and query at 30,248, 82,161, and 11,659, respectively.

MARS [238] and LS-VID [95] are two public datasets for video-based re-identification. MARS [238] contains 1,261 unique identities and a total of 17,503 tracklets, with 625 identities reserved for training and 636 for testing. LS-VID [95] includes 842 identities for training and 2,730 for testing, with training sequences comprising 8,298 video segments, while the gallery and query sets contain 11,310 and 1,980 segments, respectively.

Implementation Details. In our pipeline, we use two shape representations for guidance: PIFu [152] for both videos and frames, and SMPL [119] for videos. Since appearance is largely preserved in images, we utilize the pretrained PIFu surface model [152] for shape guidance, accompanied by a pretrained DeepLab-v3 [16] to remove the background and provide the body mask. For SMPL [119], we follow SPIN [92] to extract the 3-D body shapes for each frame. The number of points used for training, k, is set to 500. We empirically set λ_1 as 1 for both cases and λ_2 as 0.001 for video-based re-identification. During inference, we calculate the pairwise cosine similarity between the query and gallery examples for both videos and images.

For frame-based person re-identification, we resize the input images to 256×128 and use a ResNet-50 [67] to encode the features into $\boldsymbol{F}^{map}$. The shape extractor, which follows the feature map, includes 2 `UpConv` operations to upscale the feature map to its pixel-level implicit representation feature map $\boldsymbol{F}^{P}$. Each `UpConv` operation increases the feature dimension by 2 on both height and width. During training, we incorporate a Triplet loss $\mathcal{L}_{triplet}$ [158] with a 0.3 margin, along with a Cross Entropy loss $\mathcal{L}_{CE}$ as

$$\mathcal{L}_{ID} = \mathcal{L}_{triplet} + \mathcal{L}_{CE}. \tag{8.6}$$

We train the model for 120 epochs, starting with a learning rate of $3.5e^{-4}$ using the Adam optimizer, and reduce it by $\frac{1}{10}$ at epoch 40 and 70.

For video-based re-identification, we follow PSTA [200] and use the ResNet-50 as our feature encoder, followed by a three-layer pyramid for spatial and temporal attention extraction and aggregation. We include a Triplet loss $\mathcal{L}_{triplet}$ [158] with 0.3 as margin along with a Cross Entropy loss $\mathcal{L}_{CE}$ as Equation 8.6. We train the model for 500 epochs with the Adam optimizer, with an initial learning rate set at $3.5e^{-4}$. We reduce the learning rate by 0.3 at epochs 70, 140, 210, 310, and 410. During training, we randomly select 8 frames from each clip. In the inference phase for MARS [238], we select 8 frames per video, starting with the first frame and proceeding at intervals of 8 frames. If a video's length doesn't permit an 8-frame interval, we halve the interval until 8 frames can be selected. For LS-VID, we utilize all available frames in the video and average the features from the last layer of PSTA as its representation.

We conduct all our experiments on a machine equipped with 2 A100 GPUs, and recommend using 1 GPU with 24GB memory for training frame-based datasets and 2 for video-based datasets. The training time for frame-based re-identification ranges from 12 to 30 hours; for video-based re-identification, it varies between one to three days, depending on the datasets used and the I/O speed.

Baseline Methods. For frame-based re-identification, we build SEAS on a ResNet-50 encoder and compare it with multiple state-of-the-art methods, including earlier ReID methods [31, 19], recent methos that are transformer-based approaches [33, 68, 261, 260, 18], and others [223, 133, 147, 195, 229, 94, 58, 232]. In addition, we compare with some other re-identification methods [111, 15] that use 3-D body shape in addition to appearance. We assess the methods based on rank-1 accuracy and mean average precision (mAP) for comparison.

Table 8.1: Rank-1 accuracy and mAP on Market1501 [239] and MSMT17 [203] datasets. We bold the numbers that are the best performance and underline the second best ones. Methods ending with (*) include 3-D body shapes in addition to appearance.

Method	Market1501 [239]		MSMT17 [203]	
	Rank-1	mAP	Rank-1	mAP
ViT-B [33]	94.0	87.6	82.8	63.6
TransReID [68]	95.2	89.5	86.2	69.4
AAFormer [261]	95.4	87.7	83.6	63.2
AGW [223]	95.5	89.5	81.2	59.7
FlipReID [133]	95.5	89.6	85.6	68.0
CAL [147]	95.5	89.5	84.2	64.0
PFD [195]	95.5	89.7	83.8	64.4
SAN [84]	96.1	88.0	79.2	55.7
LDS [229]	95.8	90.4	86.5	67.2
DiP [94]	95.8	90.3	87.3	71.8
MPN [31]	96.4	90.1	83.5	62.7
MSINet [58]	95.3	89.6	81.0	59.6
SCSN [19]	95.7	88.5	83.8	58.5
PHA [232]	96.1	90.2	86.1	68.9
PASS ViT-B [260]	<u>96.9</u>	93.3	89.7	74.3
SOLIDER [18]	<u>96.9</u>	<u>93.9</u>	<u>90.7</u>	<u>77.1</u>
ASSP* [15]	95.0	87.3	-	-
3DInvarReID* [111]	95.1	87.9	80.8	59.1
Baseline (ResNet-50)	94.1	83.2	73.8	47.2
SEAS (ResNet-50)	**98.6**	**98.9**	**91.7**	**93.4**

For video-based re-identification, we compare with several state-of-the-art models, including models with 2-D convolutions [116, 69, 41, 75, 113, 89, 200], 3-D convolutions [74, 60, 73, 2], and ViT-based methods [33, 210, 14, 33], measuring rank-1 accuracy and mAP. Following PSTA [200], we use a ResNet-50 [67] as our per-frame image encoder, followed by the pyramid spatial-temporal aggregation.

8.3.2 Results and Analysis

Results for Frame-based Re-ID. We present the numerical results for frame-based re-identification in Table 8.1, comparing the application of SEAS on ResNet-50 with other state-of-the-art methods on Market1501 [194] and MSMT17 [203] in terms of rank-1 accuracy and mean average precision (mAP). Using

Table 8.2: Rank-1 accuracy and mAP on MARS [238] and LS-VID [95] datasets. In SEAS, we use ResNet-50 for identity feature map extraction and PSTA [200] for temporal fusion.

Method	MARS [238]		LS-VID [95]	
	Rank-1	mAP	Rank-1	mAP
GRL [116]	91.0	84.8	-	-
TokenShift [14]	90.2	86.6	80.4	68.7
ViT [33]	89.7	86.4	85.3	76.4
TCLNet [74]	89.8	85.1	-	-
AP3D [60]	90.1	85.1	-	-
DenseIL [69]	90.8	87.0	-	-
STMN [41]	90.5	84.5	82.1	69.2
BiCnet-TKS [73]	90.2	86.0	84.6	75.1
STRF [2]	90.3	86.1	-	-
RFCnet [75]	90.7	86.3	-	-
CTL [113]	91.4	86.7	-	-
DSANet [89]	91.1	86.6	85.1	75.5
CAViT [210]	90.8	87.2	89.2	79.2
Baseline (PSTA) [200]	91.5	85.8	77.7	67.2
SEAS (PSTA)	**95.1**	**96.6**	**90.5**	**93.4**

SEAS with ResNet-50 as the image encoder outperforms all other methods on both datasets across these metrics. On Market1501 [194], ResNet-50 with SEAS achieves a rank-1 accuracy of 98.6% and an mAP of 98.9%, surpassing SOLIDER [18], which had the previous highest rank-1 accuracy of 96.9% with external training data. We also observe significant improvements on MSMT17 [203], with rank-1 accuracy increasing from 90.7% to 91.7% and mAP from 77.1% to 93.4%.

Moreover, we also include a comparison with explicit appearance reconstruction as in SAN [84] and other methods [111, 15] using body shapes. SAN includes a decoder-like structure for explicitly reconstructing the appearance of the entire body, even when occlusions are present. We note that using SEAS outperforms SAN on both datasets; the gap is more significant on MSMT17 [203], where occlusions are more common, making the reconstruction of the entire body appearance impractical. Compared with other methods that use features of 3-D body shapes [111, 15] for identification, ResNet-50 with SEAS also shows significant improvement. The use of a body shape extractor with pixel-level body shape guidance

Table 8.3: Model variations analysis, with ResNet-50 as the baseline for Market1501 and PSTA [200] for MARS.

	Method	Rank-1	mAP	Params	FLOPs
(I) Appearance	Baseline (Market1501)	94.1	83.2	23.51M	4.07G
(II) Body shape as input	+ PIFu as 2^{nd} branch	94.1 (+0.0)	84.8 (+1.6)	34.80M	6.28G
	+ PIFu concatenation	94.3 (+0.2)	85.8 (+2.6)	34.89M	4.26G
(III) Body shape as supervision	+ SEAS (SPIN)	97.1 (+3.0)	97.8 (+14.6)	23.51M	4.07G
	+ SEAS (PIFu)	**98.6** (+4.5)	**98.9** (+15.7)	23.51M	4.07G
(IV) SEAS w/ calibration for video frames	Baseline (MARS)	91.5	85.8	35.43M	37.70G
	+ SEAS (w/o $\mathcal{L}_{cali.}$)	94.8 (+3.3)	96.5 (+10.7)	35.43M	37.70G
	+ SEAS (w/ $\mathcal{L}_{cali.}$)	**95.1** (+3.6)	**96.7** (+10.9)	35.43M	37.70G

establishes a stronger connection between shape and observed appearance, providing a more complete understanding for person re-identification.

Results for Video-based Re-ID. In addition to single-frame-based person re-identification, we show our results for video-based datasets [238, 95] in Table 8.2. Across both datasets, we note a significant improvement over other baseline methods. For MARS [238], the highest rank-1 and mAP from the current state-of-the-art methods are 91.5% and 87.2%, respectively. In comparison, our method achieves 95.1% for rank-1 and 96.7% for mAP, surpassing the existing baselines and reducing the errors from 8.5% to 4.9%. We also observe a performance boost on LS-VID [95]. Against the best performance for rank-1 and mAP, both from CAViT [210], which stand at 89.2% and 79.2%, using a PSTA with SEAS enhanced with $\mathcal{L}_{cali.}$ exceeds these metrics, reaching a rank-1 of 90.5% and an mAP of 93.4%. This demonstrates the efficacy of SEAS with simple encoders over other methods that leverage vision transformers.

8.3.3 Ablation Studies and Model Variations

Module Analysis. We present model components analysis in Table 8.3, comparing performance and computational costs with other variations on Market1501 [194] and MARS [238]. Based on the input and

Table 8.4: Results for applying SEAS on BoT [121] and LDS [229].

Method	Market1501 [239]		MSMT17 [203]	
	Rank-1	mAP	Rank-1	mAP
BoT [121]	94.5	85.9	74.1	50.2
w/ SEAS	**95.9**	**97.5**	**81.3**	**86.2**
LDS [229]	95.8	90.4	86.5	67.2
w/ SEAS	**96.3**	**97.8**	**86.6**	**90.1**

different ways of aggregating the body shape feature, we split the table into four categories and have the following observations:

Enhancing Appearance with Body Shape. Compared to the baseline method that relies solely on appearance in (I), incorporating body shape in (II) and (III) generally demonstrates better performance. We experiment with three variations: 1) encoding the PIFu feature with a secondary branch encoder (ResNet-18-like) and concatenating it with the appearance feature, 2) channel-wise concatenation of PIFu features with appearance as input, and 3) employing body shape as supervision following SEAS. All variations result in improved performance, confirming that the body shape feature enhances the appearance-based method.

Using Shape as Input vs. Supervision. Compared to using body shape as input to identify the person in (II), using body shape as supervision in (III) shows better performance for both metrics. While employing body shape as input incorporates general shape information, utilizing it as supervision establishes a stronger connection between pixel-level aligned body shape and appearance, as it guides the encoder to extract shape-related information based on the appearance, facilitating a more coherent integration.

Pixel and Image-level Supervision. We include experiments comparing the use of SMPL features generated by SPIN [92] and PIFu features as supervision in (III). SMPL features are image-level, and we use the reconstructed body shape β for supervision. We observe that using either of them as supervision enhances performance, while employing PIFu for pixel-level supervision shows the best results. Compared

Table 8.5: Results for applying SEAS on STMN [41] and BiCnet-TKS [73] for video-based person re-identification.

Method	MARS [238]		LS-VID [95]	
	Rank-1	mAP	Rank-1	mAP
STMN [41]	90.5	84.5	82.1	69.2
w/ SEAS	**92.2**	**94.9**	**84.1**	**88.9**
BiCnet-TKS [73]	**90.2**	86.0	84.6	75.1
w/ SEAS	90.1	**87.9**	**86.7**	**90.8**

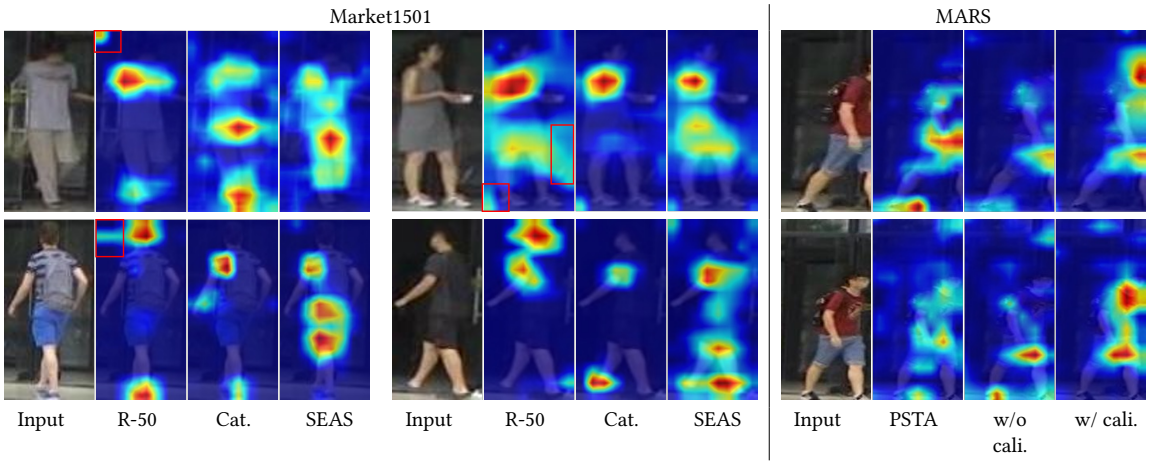


Figure 8.4: GradCam visualizations on Market1501 and MARS. For Market1501, we compare the baseline ResNet-50 (R-50) with using PIFu as input through concatenation (Cat.) and using SEAS. For MARS, we compare baseline PSTA with SEAS without and with calibration.

with image-level supervision, pixel-level supervision provides a more specific guidance between pixel-level features, making the prediction more accurate.

Cross-frame Consistency in Video. For video-based re-identification, as shown in (IV), applying PIFu as single-frame body shape guidance leads to improvements compared to the baseline method [200]. Moreover, pixel-level calibration across frames further enhances performance.

***Time Consumption**.** Since SEAS is employed only during training, its introduction in (III) and (IV) does not increase the number of parameters and FLOPs required. Conversely, explicitly using the body shape as input in (II) significantly increases these two metrics.

Generalizability. SEAS integrates an extractor with an existing identity encoder, suggesting that it could be adaptable to other methods. To validate its generalizability, we incorporate SEAS with other frame-based re-identification methods such as BoT [121] and LDS [229], as well as video-based person re-identification models including STMN [41] and BiCnet-TKS [73]. We present results comparing models with and without SEAS in Tables 8.4 and 8.5, and observe improvements on both datasets for these settings and metrics, indicating that incorporating SEAS generally enhances performance compared to the original methods. This underscores the value of incorporating the 3-D body shape extractor to enhance the identity encoder and emphasize SEAS's broad adaptability and generalizability to other methods.

Visualization. We present attention maps overlayed with input images in Figure 8.4, verifying where the model is looking at when making the decision. For Market1501, we compare the baseline with using PIFu as input and SEAS. The baseline ResNet-50, lacking body shape guidance, often erroneously directs attention to the background, as highlighted by the areas within red boxes. Introducing PIFu as input helps focus on the body by delineating its boundary, yet attention remains confined to a narrow area. SEAS shows a more evenly distributed attention across the body. As pixel-level body shape features are predicted from pixel-level appearances, using aligned body shapes as supervision ensures the appearance features related to the body region go deeper into the network and enrich the feature map with information across all body-present regions, enhancing the identity feature vector. Furthermore, we assess PSTA against SEAS, with and without calibration, on two frames of the same video in MARS. Due to the Sigmoid-based attention, PSTA scatters attention across frames and focuses narrowly on each frame. SEAS with

*For the calculation of FLOPs and the number of parameters, we refer to https://github.com/facebookresearch/fvcore/blob/main/docs/flop_count.md.

Figure 8.5: Two visualization examples from MSMT17 for SPIN (second column) and PIFu (third column) reconstructions.

calibration, however, generates a more expansive attention map, demonstrating its capability in providing enriched and consistent features.

We also visualize the extracted body shapes using examples from MSMT17 [203] in Figure 8.5. Each example displays the original image on the left, the SMPL from SPIN [92] in the middle, and the PIFu [152] shape reconstruction using the extractor on the right. The SMPL body shape from SPIN aligns well with the overall shape of the person, showing the capacity of providing dense correspondence on feature maps $\{\boldsymbol{F}_i^{map}\}$ with lower resolution. PIFu reconstructions, while limited by the image quality in re-identification datasets, effectively outline body shapes with its depth and contour, showing it is helpful to guide the identity feature encoder with pixel-level shape representations.

8.4 Conclusion and Limitation

In this work, we introduce SEAS, using Shape-Aligned Supervision, to enhance the appearance feature for person re-identification. We utilize implicit body shape representations to supervise the training of the appearance-based identity encoder, with a shape extractor to translate the feature map into pixel-level body shapes, providing pixel-level shape guidance. For video-based person re-identification, we also incorporate temporal consistency across the appearance of different frames within the same video by

adding pixel-level calibrations. Our method achieves state-of-the-art performance in both frame-based and video-based person re-identification evaluations on public datasets.

Limitations. The effectiveness of our proposed extractor relies on a pretrained shape encoder for supervision, meaning the overall performance improvement correlates with the quality of the extracted body shape features.

Chapter 9

Conclusions

In this thesis, we had a detailed analysis for different levels of mulitmodal person identification using gait and appearance. We start from the different representations of gait and analyze the introduction of 3-D body shape to enhance the gait and appearance. Since the performance of the 3-D representation can have direct impact to the recognition accuracy, we discuss two different possible improvement for the 3-D representations and apply such pixel-level implicit representations to person re-identification and enhance the appearance.

In Chapter 3, we introduce *GaitSTR*, building on GaitMix and GaitRef [254], to integrate and refine skeletons with silhouettes for gait recognition. *GaitSTR* incorporates bone representation alongside the joints, emphasizing the numerical connectivities between different nodes. It combins silhouettes and skeletons with two levels of refinement: silhouette-to-skeleton refinement for general guidance and dual-layer cross-modal adapters for sequential two-stream refinement between the joints and bones, ensuring temporal consistency across different representations. We compare *GaitSTR* on four public datasets, including CASIA-B, OUMVLP, Gait3D, and GREW, and demonstrate state-of-the-art performance compare with other gait recognition methods.

In Chapter 4, we propose the exploitation of inferring 3-D body shape from gait sequence to disentangle gait motion from appearance variances of 2-D images. In addition to the gait pattern analysis, we distill

the 3-D body shape features from selected RGB frames and transfer them to gait sequences via feature exchanging between neighbor frames. We assess our method with four state-of-the-art gait recognition methods and show better results on two public datasets at both seen and novel camera viewpoints.

In Chapter 5, we introduce ShARc, a shape and appearance-based method for identification in-the-wild. Our approach explicitly explores the contribution of body shape and appearance to the model with two encoders, pose and shape encoder for body shape and motion, and aggregated appearance encoder for human appearance. ShARc is able to handle most of the challenges for identification in the wild, such as occlusion, non-walking sequences, change of clothes, and image degradations. We have compared our method on three public datasets, including BRIAR, CCVID, and MEVID, and show state-of-the-art performance.

In Chapter 6, we have proposed Curriculum DeepSDF by designing a shape curriculum for shape representation learning. Inspired by the learning principle of humans, we organize the learning task into a series of difficulty levels from surface accuracy and sample difficulty. For surface accuracy, we design a tolerance parameter to control the global smoothness, which gradually increases the accuracy of the learned shape with more layers. For sample difficulty, we define hard, semi-hard and easy training samples in SDF learning, and gradually re-weight the samples to focus more and more on difficult local details. Experimental results show that our method largely improves the performance of DeepSDF with the same training data, training epochs and network architecture.

In Chapter 7, we introduce CaesarNeRF, a few-shot and generalizable NeRF pipeline that combines scene-level semantic with per-pixel feature representations, aiding in rendering from novel camera positions with limited reference views. We calibrate the semantic representations across different input views and employ a sequential refinement network to offer distinct semantic representations at various levels. Our method has been extensively evaluated on a broad range of datasets, exhibiting state-of-the-art performance in both generalizable and single-scene settings.

In Chapter 8, we introduce SEAS, using Shape-Aligned Supervision, to enhance the appearance feature for person re-identification. We utilize implicit body shape representations to supervise the training of the appearance-based identity encoder, with a shape extractor to translate the feature map into pixel-level body shapes, providing pixel-level shape guidance. For video-based person re-identification, we also incorporate temporal consistency across the appearance of different frames within the same video by adding pixel-level calibrations. Our method achieves state-of-the-art performance in both frame-based and video-based person re-identification evaluations on public datasets.

Following this stream, we start with using the 3-D body shape representation to enhance the identification of a person, and discuss how to improve the quality of 3-D representation and use the refined body shapes. We are excited to see the introduction of generalizable 3-D body shape and its enhancement on the input images and videos to assist recognizing the person's identity.

While the introduction of 3-D body shape significantly enhance the performance of re-identification for both gait and appearance, we notice that this is still a first step. In our pipeline, we still first reconstruct the body shape and use the pretrained model for the second step of training. Using a pretrained model can provide extra knowledge, but its reconstruction may also fails to provide what is most needed to recognize the person. Inspired by this, combining the shape extraction and person identification in an end-to-end way for training is a promising future direction. In addition, while we can further improve the reconstruction of 3-D body shape, its inference with good generalizability is significantly slower than the identification network. Optimizing the network and reducing the time consumption are potential future directions. Moreover, with the development of vision-language models that mines the similarity between vision and language properties [243, 244, 258, 251, 149, 246], large vision [197, 231] or language models [32] are also capable of providing general knowledge for matching. We believe there are still spaces ahead of this for providing a faster and more accurate 3-D body shape inference that provides further enhancement for multimodal person re-identification.

Bibliography

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. “Learning Representations and Generative Models for 3D Point Clouds”. In: *ICML*. 2018, pp. 40–49.

[2] Abhishek Aich, Meng Zheng, Srikrishna Karanam, Terrence Chen, Amit K Roy-Chowdhury, and Ziyan Wu. “Spatio-temporal representation factorization for video-based person re-identification”. In: *ICCV*. 2021, pp. 152–162.

[3] Weizhi An, Shiqi Yu, Yasushi Makihara, Xinhui Wu, Chi Xu, Yang Yu, Rijun Liao, and Yasushi Yagi. “Performance evaluation of model-based gait on multi-view very large population database with pose sequences”. In: *TBIOM* 2.4 (2020), pp. 421–430.

[4] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. “Mip-nerf 360: Unbounded anti-aliased neural radiance fields”. In: *CVPR*. 2022, pp. 5470–5479.

[5] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. “Curriculum learning”. In: *ICML*. 2009, pp. 41–48.

[6] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. “Unstructured lumigraph rendering”. In: *SIGGRAPH*. 2001, pp. 425–432.

[7] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. “OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields”. In: *TPAMI* 43.1 (2019), pp. 172–186.

[8] Jonathan C Carr, Richard K Beatson, Jon B Cherrie, Tim J Mitchell, W Richard Fright, Bruce C McCallum, and Tim R Evans. “Reconstruction and representation of 3D objects with radial basis functions”. In: *SIGGRAPH*. 2001, pp. 67–76.

[9] Tianrui Chai, Xinyu Mei, Annan Li, and Yunhong Wang. “Silhouette-Based View-Embeddings for Gait Recognition Under Multiple Views”. In: *ICIP*. 2021, pp. 2319–2323.

[10] 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 preprint arXiv:1512.03012* (2015).

[11] Xiaobin Chang, Timothy M Hospedales, and Tao Xiang. “Multi-level factorisation net for person re-identification”. In: *CVPR*. 2018, pp. 2109–2118.

[12] Hanqing Chao, Yiwei He, Junping Zhang, and Jianfeng Feng. “Gaitset: Regarding gait as a set for cross-view gait recognition”. In: *AAAI*. 2019, pp. 8126–8133.

[13] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. “Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo”. In: *ICCV*. 2021, pp. 14124–14133.

[14] Guangyi Chen, Yongming Rao, Jiwen Lu, and Jie Zhou. “Temporal coherence or temporal motion: Which is more critical for video-based person re-identification?” In: *ECCV*. 2020, pp. 660–676.

[15] Jiaxing Chen, Xinyang Jiang, Fudong Wang, Jun Zhang, Feng Zheng, Xing Sun, and Wei-Shi Zheng. “Learning 3D shape feature for texture-insensitive person re-identification”. In: *CVPR*. 2021, pp. 8146–8155.

[16] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. “Rethinking atrous convolution for semantic image segmentation”. In: *arXiv preprint arXiv:1706.05587* (2017).

[17] Peixian Chen, Wenfeng Liu, Pingyang Dai, Jianzhuang Liu, Qixiang Ye, Mingliang Xu, Qi’an Chen, and Rongrong Ji. “Occlude them all: Occlusion-aware attention network for occluded person re-id”. In: *ICCV*. 2021, pp. 11833–11842.

[18] Weihua Chen, Xianzhe Xu, Jian Jia, Hao Luo, Yaohua Wang, Fan Wang, Rong Jin, and Xiuyu Sun. “Beyond Appearance: a Semantic Controllable Self-Supervised Learning Framework for Human-Centric Visual Tasks”. In: *CVPR*. 2023.

[19] Xuesong Chen, Canmiao Fu, Yong Zhao, Feng Zheng, Jingkuan Song, Rongrong Ji, and Yi Yang. “Salience-guided cascaded suppression network for person re-identification”. In: *CVPR*. 2020, pp. 3300–3310.

[20] Yuedong Chen, Haofei Xu, Qianyi Wu, Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. “Explicit Correspondence Matching for Generalizable Neural Radiance Fields”. In: *arXiv preprint arXiv:2304.12294* (2023).

[21] Zhiqin Chen and Hao Zhang. “Learning implicit fields for generative shape modeling”. In: *CVPR*. 2019, pp. 5939–5948.

[22] Yoonki Cho, Woo Jae Kim, Seunghoon Hong, and Sung-Eui Yoon. “Part-based pseudo label refinement for unsupervised person re-identification”. In: *CVPR*. 2022, pp. 7308–7318.

[23] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. “3D-R2N2: A unified approach for single and multi-view 3d object reconstruction”. In: *ECCV*. 2016, pp. 628–644.

[24] David Cornett, Joel Brogan, Nell Barber, Deniz Aykac, Seth Baird, Nicholas Burchfield, Carl Dukes, Andrew Duncan, Regina Ferrell, Jim Goddard, et al. “Expanding Accurate Person Recognition to New Altitudes and Ranges: The BRIAR Dataset”. In: *WACV*. 2023, pp. 593–602.

[25] Brian Curless and Marc Levoy. “A volumetric method for building complex models from range images”. In: *SIGGRAPH*. 1996, pp. 303–312.

[26] Ju Dai, Pingping Zhang, Dong Wang, Huchuan Lu, and Hongyu Wang. “Video person re-identification by temporal residual learning”. In: *TIP* 28.3 (2018), pp. 1366–1377.

[27] Daniel Davila, Dawei Du, Bryon Lewis, Christopher Funk, Joseph Van Pelt, Roderic Collins, Kellie Corona, Matt Brown, Scott McCloskey, Anthony Hoogs, et al. “MEVID: Multi-view Extended Videos with Identities for Video Person Re-Identification”. In: *WACV*. 2023, pp. 1634–1643.

[28] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. “Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach”. In: *SIGGRAPH*. 1996, pp. 11–20.

[29] Congyue Deng, Chiyu “Max” Jiang, Charles R. Qi, Xinchen Yan, Yin Zhou, Leonidas Guibas, and Dragomir Anguelov. “NeRDi: Single-View NeRF Synthesis With Language-Guided Diffusion As General Image Priors”. In: *CVPR*. 2023, pp. 20637–20647.

[30] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. “Arcface: Additive angular margin loss for deep face recognition”. In: *CVPR*. 2019, pp. 4690–4699.

[31] Changxing Ding, Kan Wang, Pengfei Wang, and Dacheng Tao. “Multi-task learning with coarse priors for robust part-aware person re-identification”. In: *TPAMI* 44.3 (2020), pp. 1474–1488.

[32] Tianyu Ding, Tianyi Chen, Haidong Zhu, Jiachen Jiang, Yiqi Zhong, Jinxin Zhou, Guangzhi Wang, Zhihui Zhu, Ilya Zharkov, and Luming Liang. “The efficiency spectrum of large language models: An algorithmic survey”. In: *arXiv preprint arXiv:2312.00678* (2023).

[33] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

[34] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. “Google scanned objects: A high-quality dataset of 3d scanned household items”. In: *ICRA*. 2022, pp. 2553–2560.

[35] Yueqi Duan, Lei Chen, Jiwen Lu, and Jie Zhou. “Deep embedding learning with discriminative sampling policy”. In: *CVPR*. 2019, pp. 4964–4973.

[36] Yueqi Duan, Jiwen Lu, and Jie Zhou. “Uniformface: Learning deep equidistributed representation for face recognition”. In: *CVPR*. 2019, pp. 3415–3424.

[37] Yueqi Duan, Yu Zheng, Jiwen Lu, Jie Zhou, and Qi Tian. “Structural relational reasoning of point clouds”. In: *CVPR*. 2019, pp. 949–958.

[38] Yueqi Duan, Haidong Zhu, He Wang, Li Yi, Ram Nevatia, and Leonidas J Guibas. “Curriculum deepsdf”. In: *ECCV*. 2020, pp. 51–67.

[39] Jeffrey L Elman. “Learning and development in neural networks: The importance of starting small”. In: *Cognition* 48.1 (1993), pp. 71–99.

[40] Joshua J Engelsma, Kai Cao, and Anil K Jain. “Learning a fixed-length fingerprint representation”. In: *TPAMI* 43.6 (2019), pp. 1981–1997.

[41] Chanho Eom, Geon Lee, Junghyup Lee, and Bumsub Ham. “Video-based person re-identification with spatial and temporal memory networks”. In: *ICCV*. 2021, pp. 12036–12045.

[42] Chao Fan, Junhao Liang, Chuanfu Shen, Saihui Hou, Yongzhen Huang, and Shiqi Yu. “OpenGait: Revisiting Gait Recognition Toward Better Practicality”. In: *arXiv preprint arXiv:2211.06597* (2022).

[43] Chao Fan, Junhao Liang, Chuanfu Shen, Saihui Hou, Yongzhen Huang, and Shiqi Yu. *OpenGait: Revisiting Gait Recognition Toward Better Practicality*. 2023. arXiv: 2211.06597 [cs.CV].

[44] Chao Fan, Yunjie Peng, Chunshui Cao, Xu Liu, Saihui Hou, Jiannan Chi, Yongzhen Huang, Qing Li, and Zhiqiang He. “Gaitpart: Temporal part-based model for gait recognition”. In: *CVPR*. 2020, pp. 14225–14233.

[45] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. “K-planes: Explicit radiance fields in space, time, and appearance”. In: *CVPR*. 2023, pp. 12479–12488.

[46] Yang Fu, Ishan Misra, and Xiaolong Wang. “Multiplane NeRF-Supervised Disentanglement of Depth and Camera Pose from Videos”. In: *ICML*. 2022.

[47] Yang Fu, Xiaoyang Wang, Yunchao Wei, and Thomas Huang. “Sta: Spatial-temporal attention for large-scale video-based person re-identification”. In: *AAAI*. Vol. 33. 2019, pp. 8287–8294.

[48] Yang Fu, Yunchao Wei, Yuqian Zhou, Honghui Shi, Gao Huang, Xinchao Wang, Zhiqiang Yao, and Thomas Huang. “Horizontal pyramid matching for person re-identification”. In: *AAAI*. Vol. 33. 2019, pp. 8295–8302.

[49] Jiyang Gao and Ram Nevatia. “Revisiting temporal modeling for video-based person reid”. In: *arXiv preprint arXiv:1805.02104* (2018).

[50] Shang Gao, Jingya Wang, Huchuan Lu, and Zimo Liu. “Pose-guided visible part matching for occluded person reid”. In: *CVPR*. 2020, pp. 11744–11752.

[51] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. “Deep Structured Implicit Functions”. In: *arXiv preprint arXiv:1912.06126* (2019).

[52] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. “Learning shape templates with structured implicit functions”. In: *ICCV*. 2019, pp. 7154–7164.

[53] Angelo Genovese, Vincenzo Piuri, Konstantinos N Plataniotis, and Fabio Scotti. “PalmNet: Gabor-PCA convolutional networks for touchless palmprint recognition”. In: *TIFS* 14.12 (2019), pp. 3160–3174.

[54] Akash Godbole, Steven A Grosz, Karthik Nandakumar, and Anil K Jain. “On demographic bias in fingerprint recognition”. In: *IJCB*. 2022, pp. 1–10.

[55] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. “Implicit Geometric Regularization for Learning Shapes”. In: *arXiv preprint arXiv:2002.10099* (2020).

[56] Steven A Grosz, Joshua J Engelsma, Eryun Liu, and Anil K Jain. “C2cl: Contact to contactless fingerprint matching”. In: *TIFS* 17 (2021), pp. 196–210.

[57] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. “A papier-mâché approach to learning 3D surface generation”. In: *CVPR*. 2018, pp. 216–224.

[58] Jianyang Gu, Kai Wang, Hao Luo, Chen Chen, Wei Jiang, Yuqiang Fang, Shanghang Zhang, Yang You, and Jian Zhao. “MSINet: Twins Contrastive Search of Multi-Scale Interaction for Object ReID”. In: *CVPR*. 2023, pp. 19243–19253.

[59] Xinqian Gu, Hong Chang, Bingpeng Ma, Shutao Bai, Shiguang Shan, and Xilin Chen. “Clothes-changing person re-identification with RGB modality only”. In: *CVPR*. 2022, pp. 1060–1069.

[60] Xinqian Gu, Hong Chang, Bingpeng Ma, Hongkai Zhang, and Xilin Chen. “Appearance-preserving 3d convolution for video-based person re-identification”. In: *ECCV*. Springer. 2020, pp. 228–243.

[61] Kan Guo, Dongqing Zou, and Xiaowu Chen. “3D mesh labeling via deep convolutional neural networks”. In: *TOG* 35.1 (2015), pp. 1–12.

[62] Yuxiang Guo, Cheng Peng, Chun Pong Lau, and Rama Chellappa. “Multi-Modal Human Authentication Using Silhouettes, Gait and RGB”. In: *arXiv preprint arXiv:2210.04050* (2022).

[63] Zhenhua Guo, David Zhang, Lei Zhang, and Wangmeng Zuo. “Palmprint verification using binary orientation co-occurrence vector”. In: *PRL* 30.13 (2009), pp. 1219–1227.

[64] Guy Hacohen and Daphna Weinshall. “On The Power of Curriculum Learning in Training Deep Networks”. In: *ICML*. 2019, pp. 2535–2544.

[65] Ke Han, Shaogang Gong, Yan Huang, Liang Wang, and Tieniu Tan. “Clothing-Change Feature Augmentation for Person Re-Identification”. In: *CVPR*. 2023, pp. 22066–22075.

[66] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. “Meshcnn: a network with an edge”. In: *ACM Transactions on Graphics (ToG)* 38.4 (2019), pp. 1–12.

[67] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *CVPR*. 2016, pp. 770–778.

[68] Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. “Transreid: Transformer-based object re-identification”. In: *ICCV*. 2021, pp. 15013–15022.

[69] Tianyu He, Xin Jin, Xu Shen, Jianqiang Huang, Zhibo Chen, and Xian-Sheng Hua. “Dense interaction learning for video-based person re-identification”. In: *ICCV*. 2021, pp. 1490–1501.

[70] Tianyu He, Xu Shen, Jianqiang Huang, Zhibo Chen, and Xian-Sheng Hua. “Partial person re-identification with part-part correspondence learning”. In: *CVPR*. 2021, pp. 9105–9115.

[71] Yiwei He, Junping Zhang, Hongming Shan, and Liang Wang. “Multi-task GANs for view-specific feature learning in gait recognition”. In: *TIFS* 14.1 (2018), pp. 102–113.

[72] Peixian Hong, Tao Wu, Ancong Wu, Xintong Han, and Wei-Shi Zheng. “Fine-grained shape-appearance mutual learning for cloth-changing person re-identification”. In: *CVPR*. 2021, pp. 10513–10522.

[73] Ruibing Hou, Hong Chang, Bingpeng Ma, Rui Huang, and Shiguang Shan. “Bicnet-tks: Learning efficient spatial-temporal representation for video person re-identification”. In: *CVPR*. 2021, pp. 2014–2023.

[74] Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. “Temporal complementary learning for video person re-identification”. In: *ECCV*. Springer. 2020, pp. 388–405.

[75] Ruibing Hou, Bingpeng Ma, Hong Chang, Xinqian Gu, Shiguang Shan, and Xilin Chen. “Feature completion for occluded person re-identification”. In: *TPAMI* 44.9 (2021), pp. 4894–4912.

[76] Saihui Hou, Chunshui Cao, Xu Liu, and Yongzhen Huang. “Gait lateral network: Learning discriminative and compact representations for gait recognition”. In: *ECCV*. 2020, pp. 382–398.

[77] Shoukang Hu, Fangzhou Hong, Liang Pan, Haiyi Mei, Lei Yang, and Ziwei Liu. “SHERF: Generalizable Human NeRF from a Single Image”. In: *ICCV*. 2023, pp. 9352–9364.

[78] Xiaohu Huang, Duowang Zhu, Hao Wang, Xinggang Wang, Bo Yang, Botao He, Wenyu Liu, and Bin Feng. “Context-sensitive temporal feature learning for gait recognition”. In: *ICCV*. 2021, pp. 12909–12918.

[79] 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: *TPAMI* 36.7 (2013), pp. 1325–1339.

[80] Muhammad Zubair Irshad, Sergey Zakharov, Katherine Liu, Vitor Guizilini, Thomas Kollar, Adrien Gaidon, Zsolt Kira, and Rares Ambrus. “NeO 360: Neural Fields for Sparse View Synthesis of Outdoor Scenes”. In: *ICCV*. 2023, pp. 9187–9198.

[81] Wei Jia, De-Shuang Huang, and David Zhang. “Palmprint verification based on robust line orientation code”. In: *PR* 41.5 (2008), pp. 1504–1513.

[82] Yifan Jiang, Peter Hedman, Ben Mildenhall, Dejia Xu, Jonathan T Barron, Zhangyang Wang, and Tianfan Xue. “AligNeRF: High-Fidelity Neural Radiance Fields via Alignment-Aware Training”. In: *CVPR*. 2023, pp. 46–55.

[83] Xin Jin, Tianyu He, Kecheng Zheng, Zhiheng Yin, Xu Shen, Zhen Huang, Ruoyu Feng, Jianqiang Huang, Zhibo Chen, and Xian-Sheng Hua. “Cloth-changing person re-identification from a single image with gait prediction and regularization”. In: *CVPR*. 2022, pp. 14278–14287.

[84] Xin Jin, Cuiling Lan, Wenjun Zeng, Guoqiang Wei, and Zhibo Chen. “Semantics-aligned representation learning for person re-identification”. In: *AAAI*. Vol. 34. 07. 2020, pp. 11173–11180.

[85] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. “Geonerf: Generalizing nerf with geometry priors”. In: *CVPR*. 2022, pp. 18365–18375.

[86] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. “Perceptual losses for real-time style transfer and super-resolution”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer. 2016, pp. 694–711.

[87] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. “Progressive growing of gans for improved quality, stability, and variation”. In: *arXiv preprint arXiv:1710.10196* (2017).

[88] Minchul Kim, Anil K Jain, and Xiaoming Liu. “Adaface: Quality adaptive margin for face recognition”. In: *CVPR*. 2022, pp. 18750–18759.

[89] Minjung Kim, MyeongAh Cho, and Sangyoun Lee. “Feature Disentanglement Learning with Switching and Aggregation for Video-based Person Re-Identification”. In: *WACV*. 2023, pp. 1603–1612.

[90] Minsu Kim, Seungryong Kim, Jungin Park, Seongheon Park, and Kwanghoon Sohn. “PartMix: Regularization Strategy to Learn Part Discovery for Visible-Infrared Person Re-identification”. In: *CVPR*. 2023, pp. 18621–18632.

[91] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[92] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. “Learning to reconstruct 3D human pose and shape via model-fitting in the loop”. In: *ICCV*. 2019, pp. 2252–2261.

[93] Qingming Leng, Mang Ye, and Qi Tian. “A survey of open-world person re-identification”. In: *TCSVT* 30.4 (2019), pp. 1092–1108.

[94] Dengjie Li, Siyu Chen, Yujie Zhong, Fan Liang, and Lin Ma. “Dip: Learning discriminative implicit parts for person re-identification”. In: *arXiv preprint arXiv:2212.13906* (2022).

[95] Jianing Li, Jingdong Wang, Qi Tian, Wen Gao, and Shiliang Zhang. “Global-local temporal representations for video person re-identification”. In: *ICCV*. 2019, pp. 3958–3967.

[96] Shuang Li, Slawomir Bak, Peter Carr, and Xiaogang Wang. “Diversity regularized spatiotemporal attention for video-based person re-identification”. In: *CVPR*. 2018, pp. 369–378.

[97] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. “Deepreid: Deep filter pairing neural network for person re-identification”. In: *CVPR*. 2014, pp. 152–159.

[98] Xiang Li, Yasushi Makihara, Chi Xu, and Yasushi Yagi. “End-to-end Model-based Gait Recognition using Synchronized Multi-view Pose Constraint”. In: *ICCV*. 2021, pp. 4106–4115.

[99] Xiang Li, Yasushi Makihara, Chi Xu, Yasushi Yagi, Shiqi Yu, and Mingwu Ren. “End-to-end model-based gait recognition”. In: *ACCV*. 2020.

[100] Yulin Li, Jianfeng He, Tianzhu Zhang, Xiang Liu, Yongdong Zhang, and Feng Wu. “Diverse part discovery: Occluded person re-identification with part-aware transformer”. In: *CVPR*. 2021, pp. 2898–2907.

[101] Zhen Li, Shiyu Chang, Feng Liang, Thomas S Huang, Liangliang Cao, and John R Smith. “Learning locally-adaptive decision functions for person verification”. In: *CVPR*. 2013, pp. 3610–3617.

[102] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. “Dynibar: Neural dynamic image-based rendering”. In: *CVPR*. 2023, pp. 4273–4284.

[103] Junhao Liang, Chao Fan, Saihui Hou, Chuanfu Shen, Yongzhen Huang, and Shiqi Yu. “GaitEdge: Beyond Plain End-to-end Gait Recognition for Better Practicality”. In: *arXiv preprint arXiv:2203.03972* (2022).

[104] Rijun Liao, Shiqi Yu, Weizhi An, and Yongzhen Huang. “A model-based gait recognition method with body pose and human prior knowledge”. In: *PR* (2020).

[105] Yiyi Liao, Simon Donne, and Andreas Geiger. “Deep marching cubes: Learning explicit surface representations”. In: *CVPR*. 2018, pp. 2916–2925.

[106] Beibei Lin, Shunli Zhang, and Xin Yu. “Gait Recognition via Effective Global-Local Feature Representation and Local Temporal Aggregation”. In: *ICCV*. 2021, pp. 14648–14656.

[107] Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. “Efficient neural radiance fields for interactive free-viewpoint video”. In: *SIGGRAPH Asia 2022 Conference Papers*. 2022, pp. 1–9.

[108] Ji Lin, Chuang Gan, and Song Han. “Tsm: Temporal shift module for efficient video understanding”. In: *ICCV*. 2019, pp. 7083–7093.

[109] 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: *ECCV*. 2014, pp. 740–755.

[110] Chih-Ting Liu, Chih-Wei Wu, Yu-Chiang Frank Wang, and Shao-Yi Chien. “Spatially and temporally efficient non-local attention network for video-based person re-identification”. In: *arXiv preprint arXiv:1908.01683* (2019).

[111] Feng Liu, Minchul Kim, ZiAng Gu, Anil Jian, and Xiaoming Liu. “Learning Clothing and Pose Invariant 3D Shape Representation for Long-Term Person Re-Identification”. In: *ICCV* (2023).

[112] Hao Liu, Zequn Jie, Karlekar Jayashree, Meibin Qi, Jianguo Jiang, Shuicheng Yan, and Jiashi Feng. “Video-based person re-identification with accumulative motion context”. In: *TCSVT* 28.10 (2017), pp. 2788–2802.

[113] Jiawei Liu, Zheng-Jun Zha, Wei Wu, Kecheng Zheng, and Qibin Sun. “Spatial-temporal correlation and topology learning for person re-identification in videos”. In: *CVPR*. 2021, pp. 4370–4379.

[114] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. “Neural actor: Neural free-view synthesis of human actors with pose control”. In: *TOC* 40.6 (2021), pp. 1–16.

[115] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. “Learning to infer implicit surfaces without 3D supervision”. In: *NeurIPS*. 2019, pp. 8293–8304.

[116] Xuehu Liu, Pingping Zhang, Chenyang Yu, Huchuan Lu, and Xiaoyun Yang. “Watching you: Global-guided reciprocal learning for video-based person re-identification”. In: *CVPR*. 2021, pp. 13334–13343.

[117] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. “Neural rays for occlusion-aware image-based rendering”. In: *CVPR*. 2022, pp. 7824–7833.

[118] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. “Disentangling and unifying graph convolutions for skeleton-based action recognition”. In: *CVPR*. 2020, pp. 143–152.

[119] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. “SMPL: A skinned multi-person linear model”. In: *TOG* 34.6 (2015), pp. 1–16.

[120] William E Lorensen and Harvey E Cline. “Marching cubes: A high resolution 3D surface construction algorithm”. In: *SIGGRAPH* (1987), pp. 163–169.

[121] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. “Bag of tricks and a strong baseline for deep person re-identification”. In: *CVPR workshops*. 2019.

[122] Daniel Maturana and Sebastian Scherer. “Voxnet: A 3D convolutional neural network for real-time object recognition”. In: *IROS*. 2015, pp. 922–928.

[123] Niall McLaughlin, Jesus Martinez Del Rincon, and Paul Miller. “Recurrent convolutional network for video-based person re-identification”. In: *CVPR*. 2016, pp. 1325–1334.

[124] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. “Monocular 3d human pose estimation in the wild using improved cnn supervision”. In: *3DV*. 2017, pp. 506–516.

[125] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. “Occupancy networks: Learning 3D reconstruction in function space”. In: *CVPR*. 2019, pp. 4460–4470.

[126] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. “Deep level sets: Implicit surface representations for 3D shape inference”. In: *arXiv preprint arXiv:1901.06802* (2019).

[127] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines". In: *TOG* 38.4 (2019), pp. 1–14.

[128] 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: *ECCV*. 2020, pp. 405–421.

[129] 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: *Communications of the ACM* 65.1 (2021), pp. 99–106.

[130] Gyeongsik Moon, Hyeongjin Nam, Takaaki Shiratori, and Kyoung Mu Lee. "3d clothed human reconstruction in the wild". In: *ECCV*. 2022, pp. 184–200.

[131] Chrisopher B Nalty, Neehar Peri, Joshua Gleason, Carlos D Castillo, Shuowen Hu, Thirimachos Bourlai, and Rama Chellappa. "A Brief Survey on Person Recognition at a Distance". In: *arXiv preprint arXiv:2212.08969* (2022).

[132] Kien Nguyen, Clinton Fookes, Sridha Sridharan, Feng Liu, Xiaoming Liu, Arun Ross, Dana Michalski, Huy Nguyen, Debayan Deb, Mahak Kothari, et al. "AG-ReID 2023: Aerial-Ground Person Re-identification Challenge Results". In: ().

[133] Xingyang Ni and Esa Rahtu. "Flipreid: closing the gap between training and inference in person re-identification". In: *EUVIPW*. 2021, pp. 1–6.

[134] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. "Neural articulated radiance field". In: *ICCV*. 2021, pp. 5762–5772.

[135] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. "Multi-level partition of unity implicits". In: *SIGGRAPH*. 2005, pp. 173–180.

[136] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. "Deepsdf: Learning continuous signed distance functions for shape representation". In: *CVPR*. 2019, pp. 165–174.

[137] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. "Nerfies: Deformable neural radiance fields". In: *ICCV*. 2021, pp. 5865–5874.

[138] Priyank Pathak, Amir Erfan Eshratifar, and Michael Gormish. "Video person re-id: Fantastic techniques and where to find them (student abstract)". In: *AAAI*. Vol. 34. 2020, pp. 13893–13894.

[139] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. "Expressive Body Capture: 3D Hands, Face, and Body from a Single Image". In: *CVPR*. 2019.

[140] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. "Animatable Neural Radiance Fields for Human Body Modeling". In: *arXiv preprint arXiv:2105.02872* (2021).

[141] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. "Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans". In: *CVPR*. 2021, pp. 9054–9063.

[142] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. "D-nerf: Neural radiance fields for dynamic scenes". In: *CVPR*. 2021, pp. 10318–10327.

[143] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. "Pointnet: Deep learning on point sets for 3d classification and segmentation". In: *CVPR*. 2017, pp. 652–660.

[144] 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: *CVPR*. 2016, pp. 5648–5656.

[145] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space". In: *NeurIPS*. 2017, pp. 5099–5108.

[146] Haocong Rao and Chunyan Miao. "TranSG: Transformer-Based Skeleton Graph Prototype Contrastive Learning With Structure-Trajectory Prompted Reconstruction for Person Re-Identification". In: *CVPR*. 2023, pp. 22118–22128.

[147] Yongming Rao, Guangyi Chen, Jiwen Lu, and Jie Zhou. "Counterfactual attention learning for fine-grained visual categorization and re-identification". In: *ICCV*. 2021, pp. 1025–1034.

[148] Yongming Rao, Jiwen Lu, and Jie Zhou. "Global-Local Bidirectional Reasoning for Unsupervised Representation Learning of 3D Point Clouds". In: *CVPR*. 2020, pp. 5376–5385.

[149] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models". In: *CVPR*. 2022, pp. 10684–10695.

[150] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. "The earth mover's distance as a metric for image retrieval". In: *IJCV* 40.2 (2000), pp. 99–121.

[151] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115 (2015), pp. 211–252.

[152] 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: *ICCV*. 2019, pp. 2304–2314.

[153] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. "Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization". In: *CVPR*. 2020, pp. 84–93.

[154] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. “Mobilenetv2: Inverted residuals and linear bottlenecks”. In: *CVPR*. 2018, pp. 4510–4520.

[155] Terence D Sanger. “Neural network learning control of robot manipulators using gradually increasing task difficulty”. In: *IEEE transactions on Robotics and Automation* 10.3 (1994), pp. 323–333.

[156] Johannes Lutz Schönberger and Jan-Michael Frahm. “Structure-from-Motion Revisited”. In: *CVPR*. 2016.

[157] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. “Pixelwise View Selection for Unstructured Multi-View Stereo”. In: *ECCV*. 2016.

[158] Florian Schroff, Dmitry Kalenichenko, and James Philbin. “Facenet: A unified embedding for face recognition and clustering”. In: *CVPR*. 2015, pp. 815–823.

[159] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. “A comparison and evaluation of multi-view stereo reconstruction algorithms”. In: *CVPR*. 2006, pp. 519–528.

[160] 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: *ICCV*. 2017, pp. 618–626.

[161] Chen Shen, James F O’Brien, and Jonathan R Shewchuk. “Interpolating and approximating implicit surfaces from polygon soup”. In: *SIGGRAPH*. 2004, pp. 896–904.

[162] Chuanfu Shen, Chao Fan, Wei Wu, Rui Wang, George Q Huang, and Shiqi Yu. “LIDAR GAIT: Benchmarking 3D Gait Recognition with Point Clouds”. In: *arXiv preprint arXiv:2211.10598* (2022).

[163] Lei Shen, Jianlong Jin, Ruixin Zhang, Huaen Li, Kai Zhao, Yingyi Zhang, Jingyun Zhang, Shouhong Ding, Yang Zhao, and Wei Jia. “RPG-Palm: Realistic Pseudo-data Generation for Palmprint Recognition”. In: *ICCV*. 2023, pp. 19605–19616.

[164] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. “Two-stream adaptive graph convolutional networks for skeleton-based action recognition”. In: *CVPR*. 2019, pp. 12026–12035.

[165] Minho Shim, Hsuan-I Ho, Jinhyung Kim, and Dongyoon Wee. “Read: Reciprocal attention discriminator for image-to-video re-identification”. In: *ECCV*. 2020, pp. 335–350.

[166] Kohei Shiraga, Yasushi Makihara, Daigo Muramatsu, Tomio Echigo, and Yasushi Yagi. “Geinet: View-invariant gait recognition using a convolutional neural network”. In: *ICB*. 2016, pp. 1–8.

[167] Jianlou Si, Honggang Zhang, Chun-Guang Li, Jason Kuen, Xiangfei Kong, Alex C Kot, and Gang Wang. “Dual attention matching network for context-aware feature sequence based person re-identification”. In: *CVPR*. 2018, pp. 5363–5372.

[168] Ayan Sinha, Jing Bai, and Karthik Ramani. “Deep learning 3D shape surfaces using geometry images”. In: *ECCV*. 2016, pp. 223–240.

[169] Vladimir Somers, Christophe De Vleeschouwer, and Alexandre Alahi. “Body part-based representation learning for occluded person Re-Identification”. In: *WACV*. 2023, pp. 1613–1623.

[170] Chunfeng Song, Yongzhen Huang, Yan Huang, Ning Jia, and Liang Wang. “Gaitnet: An end-to-end network for gait based human identification”. In: *PR* 96 (2019), p. 106988.

[171] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. “Multi-view convolutional neural networks for 3D shape recognition”. In: *ICCV*. 2015, pp. 945–953.

[172] Arulkumar Subramaniam, Athira Nambiar, and Anurag Mittal. “Co-segmentation inspired attention networks for video-based person re-identification”. In: *ICCV*. 2019, pp. 562–572.

[173] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. “Generalizable patch-based neural rendering”. In: *ECCV*. 2022, pp. 156–174.

[174] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. “Deep high-resolution representation learning for human pose estimation”. In: *CVPR*. 2019, pp. 5693–5703.

[175] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. “Deep high-resolution representation learning for human pose estimation”. In: *CVPR*. 2019, pp. 5693–5703.

[176] Yan Sun, Xueling Feng, Liyan Ma, Long Hu, and Mark Nixon. “TriGait: Aligning and Fusing Skeleton and Silhouette Gait Data via a Tri-Branch Network”. In: *IJCB*. 2023.

[177] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Michael J Black, and Tao Mei. “Monocular, one-stage, regression of multiple 3d people”. In: *ICCV*. 2021, pp. 11179–11188.

[178] Yu Sun, Wu Liu, Qian Bao, Yili Fu, Tao Mei, and Michael J Black. “Putting people in their place: Monocular regression of 3d people in depth”. In: *CVPR*. 2022, pp. 13243–13252.

[179] Noriko Takemura, Yasushi Makihara, Daigo Muramatsu, Tomio Echigo, and Yasushi Yagi. “Multi-view large population gait dataset and its performance evaluation for cross-view gait recognition”. In: *TCVA* 10.1 (2018), pp. 1–14.

[180] Torben Teepe, Ali Khan, Johannes Gilg, Fabian Herzog, Stefan Hörmann, and Gerhard Rigoll. “Gaitgraph: Graph convolutional network for skeleton-based gait recognition”. In: *ICIP*. 2021, pp. 2314–2318.

[181] Maoqing Tian, Shuai Yi, Hongsheng Li, Shihua Li, Xuesen Zhang, Jianping Shi, Junjie Yan, and Xiaogang Wang. “Eliminating background-bias for robust person re-identification”. In: *CVPR*. 2018, pp. 5794–5803.

[182] Yonglong Tian, Dilip Krishnan, and Phillip Isola. “Contrastive representation distillation”. In: *arXiv preprint arXiv:1910.10699* (2019).

[183] Alex Trevithick and Bo Yang. “Grf: Learning a general radiance field for 3d representation and rendering”. In: *ICCV*. 2021, pp. 15182–15192.

[184] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. "Multi-view supervision for single-view reconstruction via differentiable ray consistency". In: *CVPR*. 2017, pp. 2626–2634.

[185] Greg Turk and James F O'brien. "Modelling with implicit surfaces that interpolate". In: *TOG* 21.4 (2002), pp. 855–873.

[186] Greg Turk and James F O'brien. "Shape transformation using variational implicit functions". In: *SIGGRAPH*. 1999, pp. 14–20.

[187] Mukund Varma, Peihao Wang, Xuxi Chen, Tianlong Chen, Subhashini Venugopalan, and Zhangyang Wang. "Is Attention All That NeRF Needs?" In: *ICLR*. 2023.

[188] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *NeurIPS* 30 (2017).

[189] Michael Waechter, Nils Moehrle, and Michael Goesele. "Let there be color! Large-scale texturing of 3D reconstructions". In: *ECCV*. 2014, pp. 836–850.

[190] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. "Deep high-resolution representation learning for visual recognition". In: *TPAMI* 43.10 (2020), pp. 3349–3364.

[191] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. "Pixel2mesh: Generating 3D mesh models from single RGB images". In: *ECCV*. 2018, pp. 52–67.

[192] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. "Ibrnet: Learning multi-view image-based rendering". In: *CVPR*. 2021, pp. 4690–4699.

[193] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. "Arah: Animatable volume rendering of articulated human sdfs". In: *ECCV*. 2022, pp. 1–19.

[194] Taiqing Wang, Shaogang Gong, Xiatian Zhu, and Shengjin Wang. "Person re-identification by video ranking". In: *ECCV*. 2014, pp. 688–703.

[195] Tao Wang, Hong Liu, Pinhao Song, Tianyu Guo, and Wei Shi. "Pose-guided feature disentangling for occluded person re-identification based on transformer". In: *AAAI*. Vol. 36. 3. 2022, pp. 2540–2549.

[196] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. "Rodin: A generative model for sculpting 3d digital avatars using diffusion". In: *CVPR*. 2023, pp. 4563–4573.

[197] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. "Internimage: Exploring large-scale vision foundation models with deformable convolutions". In: *CVPR*. 2023, pp. 14408–14419.

[198] Yanan Wang, Xuezhi Liang, and Shengcai Liao. "Cloning outfits from real-world images to 3D characters for generalizable person re-identification". In: *CVPR*. 2022, pp. 4900–4909.

[199] Yicheng Wang, Zhenzhong Chen, Feng Wu, and Gang Wang. “Person re-identification with cascaded pairwise convolutions”. In: *CVPR*. 2018, pp. 1470–1478.

[200] Yingquan Wang, Pingping Zhang, Shang Gao, Xia Geng, Hu Lu, and Dong Wang. “Pyramid spatial-temporal aggregation for video-based person re-identification”. In: *ICCV*. 2021, pp. 12026–12035.

[201] 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: *TOG)* 38.5 (2019), pp. 1–12.

[202] Zheng Wang, Zhixiang Wang, Yinqiang Zheng, Yang Wu, Wenjun Zeng, and Shin’ichi Satoh. “Beyond Intra-modality: A Survey of Heterogeneous Person Re-identification”. In: *IJCAI*. 2020.

[203] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. “Person transfer gan to bridge domain gap for person re-identification”. In: *CVPR*. 2018, pp. 79–88.

[204] Daphna Weinshall, Gad Cohen, and Dan Amir. “Curriculum Learning by Transfer Learning: Theory and Experiments with Deep Networks”. In: *ICML*. 2018, pp. 5238–5246.

[205] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. “A discriminative feature learning approach for deep face recognition”. In: *ECCV*. 2016, pp. 499–515.

[206] Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. “Humannerf: Free-viewpoint rendering of moving people from monocular video”. In: *CVPR*. 2022, pp. 16210–16220.

[207] Mikołaj Wieczorek, Barbara Rychalska, and Jacek Dąbrowski. “On the unreasonable effectiveness of centroids in image retrieval”. In: *ICONIP*. 2021, pp. 212–223.

[208] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. “Nex: Real-time view synthesis with neural basis expansion”. In: *CVPR*. 2021, pp. 8534–8543.

[209] Ancong Wu, Wei-Shi Zheng, Hong-Xing Yu, Shaogang Gong, and Jianhuang Lai. “RGB-infrared cross-modality person re-identification”. In: *ICCV*. 2017, pp. 5380–5389.

[210] Jinlin Wu, Lingxiao He, Wu Liu, Yang Yang, Zhen Lei, Tao Mei, and Stan Z Li. “CAViT: Contextual Alignment Vision Transformer for Video Object Re-identification”. In: *ECCV*. 2022, pp. 549–566.

[211] Yiming Wu, Omar El Farouk Bourahla, Xi Li, Fei Wu, Qi Tian, and Xue Zhou. “Adaptive graph representation learning for video person re-identification”. In: *TIP* 29 (2020), pp. 8821–8830.

[212] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. “3D ShapeNets: A deep representation for volumetric shapes”. In: *CVPR*. 2015, pp. 1912–1920.

[213] Zifeng Wu, Yongzhen Huang, Liang Wang, Xiaogang Wang, and Tieniu Tan. “A comprehensive study on cross-view gait based human identification with deep cnns”. In: *TPAMI* 39.2 (2016), pp. 209–226.

[214] Yuliang Xiu, Jinlong Yang, Xu Cao, Dimitrios Tzionas, and Michael J Black. “ECON: Explicit Clothed humans Optimized via Normal integration”. In: *CVPR*. 2023, pp. 512–523.

[215] Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J Black. “Icon: Implicit clothed humans obtained from normals”. In: *CVPR*. 2022, pp. 13286–13296.

[216] Haojun Xu, Yan Gao, Zheng Hui, Jie Li, and Xinbo Gao. “Language Knowledge-Assisted Representation Learning for Skeleton-Based Action Recognition”. In: *arXiv preprint arXiv:2305.12398* (2023).

[217] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. “DISN: Deep implicit surface network for high-quality single-view 3D reconstruction”. In: *NeurIPS*. 2019, pp. 490–500.

[218] Shuangjie Xu, Yu Cheng, Kang Gu, Yang Yang, Shiyu Chang, and Pan Zhou. “Jointly attentive spatial-temporal pooling networks for video-based person re-identification”. In: *ICCV*. 2017, pp. 4733–4742.

[219] Sijie Yan, Yuanjun Xiong, and Dahua Lin. “Spatial temporal graph convolutional networks for skeleton-based action recognition”. In: *AAAI*. 2018.

[220] Jiawei Yang, Marco Pavone, and Yue Wang. “FreeNeRF: Improving Few-shot Neural Rendering with Free Frequency Regularization”. In: *CVPR*. 2023, pp. 8254–8263.

[221] Yang Yang, Shengcai Liao, Zhen Lei, and Stan Li. “Large scale similarity learning using similar pairs for person verification”. In: *AAAI*. 2016.

[222] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. “Mvsnet: Depth inference for unstructured multi-view stereo”. In: *ECCV*. 2018, pp. 767–783.

[223] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven CH Hoi. “Deep learning for person re-identification: A survey and outlook”. In: *TPAMI* 44.6 (2021), pp. 2872–2893.

[224] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. “pixelnerf: Neural radiance fields from one or few images”. In: *CVPR*. 2021, pp. 4578–4587.

[225] Shiqi Yu, Haifeng Chen, Qing Wang, Linlin Shen, and Yongzhen Huang. “Invariant feature extraction for gait recognition using only one uniform model”. In: *Neurocomputing* 239 (2017), pp. 81–93.

[226] Shiqi Yu, Daoliang Tan, and Tieniu Tan. “A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition”. In: *ICPR*. Vol. 4. 2006, pp. 441–444.

[227] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. “Mvimgnet: A large-scale dataset of multi-view images”. In: *CVPR*. 2023, pp. 9150–9161.

[228] Xianghao Zang, Ge Li, and Wei Gao. “Multidirection and Multiscale Pyramid in Transformer for Video-Based Pedestrian Retrieval”. In: *TII* 18.12 (2022), pp. 8776–8785.

[229] Xianghao Zang, Ge Li, Wei Gao, and Xiujun Shu. “Learning to disentangle scenes for person re-identification”. In: *IVC* 116 (2021), p. 104330.

[230] Ailing Zeng, Lei Yang, Xuan Ju, Jiefeng Li, Jianyi Wang, and Qiang Xu. “Smoothnet: A plug-and-play network for refining human poses in videos”. In: *ECCV* (2022).

[231] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. “Scaling vision transformers”. In: *CVPR*. 2022, pp. 12104–12113.

[232] Guiwei Zhang, Yongfei Zhang, Tianyu Zhang, Bo Li, and Shiliang Pu. “PHA: Patch-Wise High-Frequency Augmentation for Transformer-Based Person Re-Identification”. In: *CVPR*. 2023, pp. 14133–14142.

[233] Quan Zhang, Kaiheng Dang, Jian-Huang Lai, Zhanxiang Feng, and Xiaohua Xie. “Modeling 3D layout for group re-identification”. In: *CVPR*. 2022, pp. 7512–7520.

[234] Zhizheng Zhang, Cuiling Lan, Wenjun Zeng, Xin Jin, and Zhibo Chen. “Relation-aware global attention for person re-identification”. In: *CVPR*. 2020, pp. 3186–3195.

[235] Fuqiang Zhao, Wei Yang, Jiakai Zhang, Pei Lin, Yingliang Zhang, Jingyi Yu, and Lan Xu. “Humannerf: Efficiently generated human radiance field from sparse inputs”. In: *CVPR*. 2022, pp. 7743–7753.

[236] Kai Zhao, Lei Shen, Yingyi Zhang, Chuhan Zhou, Tao Wang, Ruixin Zhang, Shouhong Ding, Wei Jia, and Wei Shen. “BézierPalm: A free lunch for palmprint recognition”. In: *ECCV*. 2022, pp. 19–36.

[237] Jinkai Zheng, Xinchen Liu, Wu Liu, Lingxiao He, Chenggang Yan, and Tao Mei. “Gait Recognition in the Wild with Dense 3D Representations and A Benchmark”. In: *CVPR*. 2022, pp. 20228–20237.

[238] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian. “Mars: A video benchmark for large-scale person re-identification”. In: *ECCV*. 2016, pp. 868–884.

[239] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. “Scalable person re-identification: A benchmark”. In: *ICCV*. 2015, pp. 1116–1124.

[240] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, Yi Yang, and Qi Tian. “Person re-identification in the wild”. In: *CVPR*. 2017, pp. 1367–1376.

[241] Meng Zheng, Srikrishna Karanam, Ziyan Wu, and Richard J Radke. “Re-identification with consistent attentive siamese networks”. In: *CVPR*. 2019, pp. 5735–5744.

[242] Wanrong Zheng, Haidong Zhu, Zhaoheng Zheng, and Ram Nevatia. “GaitSTR: Gait Recognition with Sequential Two-stream Refinement”. In: *arXiv preprint arXiv:2404.02345* (2023).

[243] Zhaoheng Zheng, Jingmin Wei, Xuefeng Hu, Haidong Zhu, and Ram Nevatia. “Large Language Models are Good Prompt Learners for Low-Shot Image Classification”. In: *CVPR*. 2024.

[244] Zhaoheng Zheng, Haidong Zhu, and Ram Nevatia. "CAILA: Concept-Aware Intra-Layer Adapters for Compositional Zero-Shot Learning". In: *WACV*. 2024, pp. 1721–1731.

[245] Dexing Zhong and Jinsong Zhu. "Centralized large margin cosine loss for open-set deep palmprint recognition". In: *TCSVT* 30.6 (2019), pp. 1559–1568.

[246] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. "Learning to prompt for vision-language models". In: *IJCV* 130.9 (2022), pp. 2337–2348.

[247] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. "Stereo magnification: Learning view synthesis using multiplane images". In: *arXiv preprint arXiv:1805.09817* (2018).

[248] Zhen Zhou, Yan Huang, Wei Wang, Liang Wang, and Tieniu Tan. "See the forest for the trees: Joint spatial and temporal recurrent neural networks for video-based person re-identification". In: *CVPR*. 2017, pp. 4747–4756.

[249] Haidong Zhu, Pranav Budhwant, Zhaoheng Zheng, and Ram Nevatia. "SEAS: Shape-Aligned Superivision for Person Re-Identification". In: *CVPR*. 2024.

[250] Haidong Zhu, Tianyu Ding, Tianyi Chen, Ilya Zharkov, Ram Nevatia, and Luming Liang. "CaesarNeRF: Calibrated Semantic Representation for Few-shot Generalizable Neural Rendering". In: *arXiv preprint arXiv:2311.15510* (2023).

[251] Haidong Zhu, Arka Sadhu, Zhaoheng Zheng, and Ram Nevatia. "Utilizing every image object for semi-supervised phrase grounding". In: *WACV*. 2021, pp. 2210–2219.

[252] Haidong Zhu, Yuyin Sun, Chi Liu, Lu Xia, Jiajia Luo, Nan Qiao, Ram Nevatia, and Cheng-Hao Kuo. "Multimodal neural radiance field". In: *ICRA*. 2023, pp. 9393–9399.

[253] Haidong Zhu, Ye Yuan, Yiheng Zhu, Xiao Yang, and Ram Nevatia. "Open: Order-preserving pointcloud encoder decoder network for body shape refinement". In: *ICPR*. 2022, pp. 521–527.

[254] Haidong Zhu, Wanrong Zheng, Zhaoheng Zheng, and Ram Nevatia. "Gaitref: Gait recognition with refined sequential skeletons". In: *IJCB*. 2023.

[255] Haidong Zhu, Wanrong Zheng, Zhaoheng Zheng, and Ram Nevatia. "ShARc: Shape and Appearance Recognition for Person Identification In-the-wild". In: *WACV*. 2024, pp. 6290–6300.

[256] Haidong Zhu, Zhaoheng Zheng, and Ram Nevatia. "Gait Recognition Using 3-D Human Body Shape Inference". In: *WACV*. 2023, pp. 909–918.

[257] Haidong Zhu, Zhaoheng Zheng, and Ram Nevatia. "Temporal Shift and Attention Modules for Graphical Skeleton Action Recognition". In: *ICPR*. 2022, pp. 3145–3151.

[258] Haidong Zhu, Zhaoheng Zheng, Mohammad Soleymani, and Ram Nevatia. "Self-supervised learning for sentiment analysis via image-text matching". In: *ICASSP*. 2022, pp. 1710–1714.

[259] Haidong Zhu, Zhaoheng Zheng, Wanrong Zheng, and Ram Nevatia. “CAT-NeRF: Constancy-Aware Tx2Former for Dynamic Body Modeling”. In: *CVPRW*. 2023, pp. 6618–6627.

[260] Kuan Zhu, Haiyun Guo, Tianyi Yan, Yousong Zhu, Jinqiao Wang, and Ming Tang. “PASS: Part-Aware Self-Supervised Pre-Training for Person Re-Identification”. In: *ECCV*. 2022, pp. 198–214.

[261] Kuan Zhu, Haiyun Guo, Shiliang Zhang, Yaowei Wang, Jing Liu, Jinqiao Wang, and Ming Tang. “Aaformer: Auto-aligned transformer for person re-identification”. In: *TNNLS* (2023).

[262] Zheng Zhu, Xianda Guo, Tian Yang, Junjie Huang, Jiankang Deng, Guan Huang, Dalong Du, Jiwen Lu, and Jie Zhou. “Gait recognition in the wild: A benchmark”. In: *ICCV*. 2021, pp. 14789–14799.

[263] Yiyu Zhuang, Hao Zhu, Xusen Sun, and Xun Cao. “Mofanerf: Morphable facial neural radiance field”. In: *ECCV*. 2022, pp. 268–285.

[264] Yang Zou, Xiaodong Yang, Zhiding Yu, BVK Vijaya Kumar, and Jan Kautz. “Joint disentangling and adaptation for cross-domain person re-identification”. In: *ECCV*. 2020, pp. 87–104.