The Supplementary Material for LidarGait++: Learning Local Features and Size Awareness from LiDAR Point Clouds for 3D Gait Recognition

Chuanfu Shen[1,2] Rui Wang[2] Lixin Duan[1,†] Shiqi Yu[2,†]

[1] Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China

[2] Southern University of Science and Technology [†]Corresponding Author

https://lidargaitv2.github.io/

A. Impact of the Training Strategies

In our experiments, we observed that training strategies have a significant impact on point-based methods, whereas projection-based methods are typically less sensitive to the choice of training strategies. This paper explores three training strategies used in LidarGait [4] (*i.e.* Strategy-L), Baseline (*i.e.* Strategy-B), and our proposed LidarGait++ (*i.e.* Strategy-L++).

We found that both projection- and point-based methods are particularly sensitive to the initial learning rate. While point-based methods face more challenging on optimization due to the sparse and unordered nature of point clouds. A larger initial learning rate helps in effectively initializing point-based models. Additionally, we observed that using a large batch size and automatically adjusting the learning rate dynamically can slightly improve the model discriminativeness. From Tab. 2, it is clear that our point-based LidarGait++ outperforms the SoTA projection-based method, LidarGait, under all training strategies.

Table 1. Comparison on three training strategies.

Setting	*Strategy-L*	*Strategy-B*	*Strategy-L++*
Batch Size (p, k)	$(8, 8)$	$(8, 8)$	$(32, 4)$
Initial Learning Rate	0.1	0.01	0.1
Learning Rate Scheduler	MultiStepLR	MultiStepLR	CosineAnnealingLR
Optimizer	SGD	SGD	SGD
Loss Functions	Triplet + CE	Triplet + CE	Triplet + CE

Table 2. Fair comparison to SoTA under same training strategies. The result of HMRNet [1] is not included because it has not been open-sourced.

Method	*Strategy-L*	*Strategy-B*	*Strategy-L++*
LidarGait	86.8%	83.5%	86.2%
LidarGait++	**90.8%**	**87.6%**	**92.7%**

B. Hierarchical Pyramid Architecture

We conducted an ablation study to evaluate the effectiveness of the hierarchical pyramid architecture within the pyramid point pooling layer. Additionally, we explored the impact of scale and bin number, as outlined in Tab. 3. The results show that hierarchical pyramid learning significantly boosts performance, particularly when using the partitions $[1, 2]$ and $[1, 2, 4, 8, 16]$, which achieve notable improvements of +0.9% and +0.2%, respectively. However, intermediate configurations such as $[1, 2, 4]$ and $[1, 2, 4, 8]$ exhibit slight performance drops, suggesting that an optimal balance in bin selection is crucial.

Table 3. Impact of hierarchical pyramid scale and bin numbers on performance in pyramid point pooling.

Scale	List of #Bin	Result (%)
1	[1]	80.8
2	[2]	86.2
3	[4]	91.3
4	[8]	92.6
5	[16]	92.5
1	[1]	80.8
2	[1, 2]	87.1 (+0.9)
3	[1, 2, 4]	90.7 (-0.6)
4	[1, 2, 4, 8]	92.2 (-0.3)
5	[1, 2, 4, 8, 16]	92.7 (+0.2)

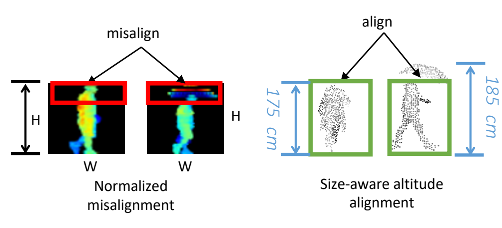

Figure 1. Illustration of alignment improvement in the umbrella subset through size-aware learning.

C. Superior Performance on Umbrella Subset

LidarGait [4] demonstrated that projection-based methods perform poorly on the umbrella subset of the SUSTech1K

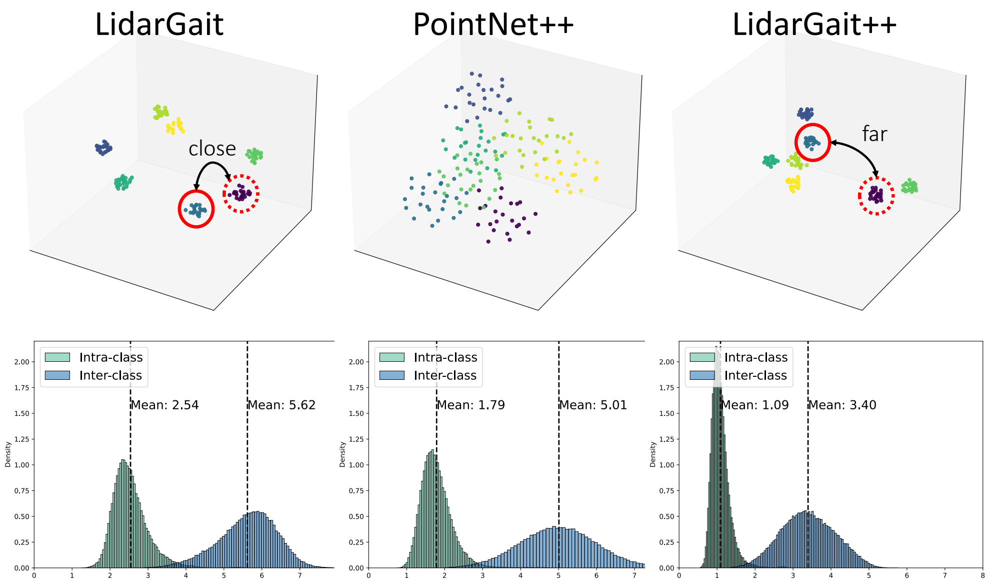

Figure 2. Qualitative visualization. Best viewed in **Zoom-in pdf**.

dataset. This limitation stems from these methods' reliance on normalization to align 2D imagery inputs, which causes misalignment between the body's head and the umbrella, as shown in Fig. 1. To address this issue, our approach introduces an end-to-end feature extraction process from point cloud data, incorporating a size-aware learning mechanism. As shown in Fig. 1, this size-aware altitude alignment ensures that all gait sequences in point cloud form are consistently aligned to the ground plane. This alignment effectively resolves the spatial misalignment issues common in 2D gait recognition by ensuring subjects with similar heights are naturally grouped together, while excluding the influence of covariates such as the umbrella. By incorporating altitude information, our method significantly enhances alignment, leading to improved performance on the umbrella subset.

D. Qualitative analysis

We visualize t-SNE and intra-class/inter-class distance in Fig. 2. With size prior, LidarGait++ minimizes intra-class distance and variance, resulting in **tighter** clustering. The blue samples are clustering with others by shape for LidarGait, while our method groups blue samples by height, eliminating the ambiguity (red circle).

E. Robustness of Size Prior

We evaluated robustness on SUSTech1K under three challenging scenarios: partial occlusion, multiple persons, and varying heights, as shown in Fig. 3. By randomly occluding partial point cloud sequences, our method showed superior performance over projection-based LidarGait as shown in Tab. 4. Our P^3 layer and size-aware learning mechanism effectively enhance robustness, addressing occlusion with size-awareness and locality.

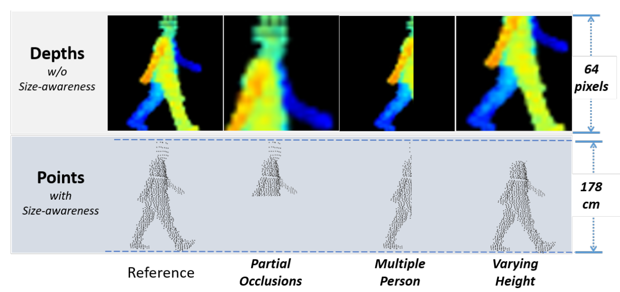

Figure 3. Robustness evaluation on SUSTech1K(%).

Table 4. Performance comparison under challenging conditions.

Model	Method	Without Occlusion	Partial Occlusion	Multiple Person	Varying Height
LidarGait	Projection-based	86.8	56.6	58.8	56.9
PointNet++ (Baseline)	Point-based	77.1	52.6	55.0	53.1
↪ LidarGait++ (Ours)	Point-based	**92.7**	**62.3**	**65.5**	**62.7**

F. Generalizability.

From Tab. 5, we can observe that all point-based models can enjoy accuracy gains consistently from our methods.

Table 5. Generalization on other models.

PointNet [2]	PointNet++ [3]	DGCNN [5]	PointTransformer [5]
32.2%(+6.9)	92.7%(+15.6)	85.5%(+33.7)	71.8%(+27.4)

References

[1] Xiao Han, Yiming Ren, Peishan Cong, Yujing Sun, Jingya Wang, Lan Xu, and Yuexin Ma. Gait recognition in large-scale free environment via single lidar. In *ACM MM*, page 380–389, 2024. 1

[2] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017. 2

[3] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 30, 2017. 2

[4] Chuanfu Shen, Chao Fan, Wei Wu, Rui Wang, George Q Huang, and Shiqi Yu. Lidargait: Benchmarking 3d gait recognition with point clouds. In *CVPR*, pages 1054–1063, 2023. 1

[5] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM TOG*, 38(5):1–12, 2019. 2