Learning by Neighbor-Aware Semantics, Deciding by Open-form Flows: Towards Robust Zero-Shot Skeleton Action Recognition

Supplementary Material

Appendix Roadmap

The supplementary material is organized into the following sections:

A. Datasets

NTU RGB+D 60 [38]. The dataset consists of 56,880 skeleton sequences spanning 60 action categories, performed by 40 subjects and captured from three distinct camera views. It has two standard evaluation protocols, including cross-subject (Xsub) and cross-view (Xview). (i) In the Xsub setting, all sequences are split according to subject identities, with 20 subjects used for training and the remaining 20 for testing. (ii) In the Xview setting, the data are divided by camera viewpoints, where view2 and view3 are used for training, and view1 is used for testing.

NTU RGB+D 120 [27]. The dataset is an extended version of the NTU RGB+D 60 [38] dataset. Compared with the former, this dataset comprises 114,480 sequences covering 120 action categories. Meanwhile, it also provides two official evaluation protocols, including the cross-subject (Xsub) and cross-setup (Xset). (i) In the Xsub setting, sequences from 53 subjects are used for training, while those from the remaining subjects are reserved for testing. (ii) In the Xset setting, data captured using cameras with even IDs are used for training, and those with odd IDs are used for testing.

PKU-MMD [7]. The dataset contains approximately 20,000 skeleton sequences across 51 action categories and is organized into two phases with progressively increasing difficulty. Specifically, it also provides two official evaluation protocols, including the cross-subject (Xsub) and cross-view (Xview). (i) In the Xsub setting, sequences from 57 subjects are used for training, while those from the remaining 9 subjects are reserved for testing. (ii) In the Xview setting, data captured from the middle and right camera views are used for training, and those from the left view are used for testing. Following [3, 4, 46], we conduct all experiments on the first phase of the dataset.

B. Dataset Seen-Unseen Split Details

Basic Seen-Unseen Split Details. Table 7 summarizes the basic seen–unseen splits used in our experiments. The 55/5 and 48/12 splits for NTU-60, as well as the 110/10 and 96/24 splits for NTU-120, follow the official settings in [11]. For PKU-MMD, we follow [3] using the 46/5 and 39/12 splits.

Table 7. Basic seen-unseen split details.

Dataset	Split Details (Unseen Category Indices)
NTU-60 [38]:	
55/5 Split [11]	[10, 11, 19, 26, 56]
48/12 Split [11]	[3, 5, 9, 12, 15, 40, 42, 47, 51, 56, 58, 59]
NTU-120 [27]:	
110/10 Split [11]	[4, 13, 37, 43, 49, 65, 88, 95, 99, 106]
96/24 Split [11]	[5, 9, 11, 16, 18, 20, 22, 29, 35, 39, 45, 49, 59, 68, 70, 81, 84, 87, 93, 94, 104, 113, 114, 119]
PKU-MMD [7]:	
46/5 Split [3]	[1, 9, 20, 34, 50]
39/12 Split [3]	[3, 7, 11, 15, 19, 21, 25, 31, 33, 36, 43, 48]

Challenging Seen-Unseen Split Details. Table 8 sum-

Table 8. Challenging seen-unseen split details.

Dataset	Split Details (Unseen Category Indices)
NTU-60 [38]:	
40/20 Split [47]	[0, 12, 13, 14, 15, 16, 17, 22, 23, 26, 29, 30, 31, 35, 36, 42, 43, 48, 56, 57]
30/30 Split [47]	[0, 1, 2, 6, 7, 8, 10, 12, 13, 15, 16, 18, 20, 21, 25, 26, 27, 31, 32, 33, 39, 42, 45, 47, 48, 51, 52, 55, 58, 59]
NTU-120 [27]:	
80/40 Split [47]	[11, 12, 18, 22, 23, 26, 28, 34, 37, 38, 42, 44, 46, 47, 48, 57, 59, 64, 66, 70, 73, 74, 75, 83, 86, 90, 92, 93, 95, 96, 102, 104, 107, 108, 110, 112, 115, 116, 118, 119]
60/60 Split [47]	[0, 1, 4, 6, 7, 8, 9, 17, 18, 21, 23, 25, 26, 28, 30, 32, 33, 34, 37, 38, 39, 40, 41, 42, 44, 45, 50, 51, 52, 53, 56, 61, 62, 65, 67, 68, 69, 70, 74, 77, 78, 81, 83, 87, 89, 90, 91, 92, 94, 95, 96, 97, 100, 101, 109, 111, 114, 115, 116, 118]

marizes the challenging seen–unseen splits used in our experiments. The 40/20 and 30/30 splits for NTU-60 and the 80/40 and 60/60 splits for NTU-120 are adopted from [47].

Random Seen-Unseen Split Details. Table 9 summarizes the three random seen–unseen splits proposed in SA-DAVE [23] and STAR-SMIE [3, 46]. The SA-DAVE benchmark is evaluated using ST-GCN features, whereas the STAR-SMIE benchmark employs Shift-GCN features. Notably, STAR-SMIE combines the STAR [3] and SMIE [46] settings, where STAR defines the PKU-MMD I random splits and SMIE defines the NTU-60 and NTU-120 random splits.

Table 9. Three random seen–unseen splits proposed by SA-DAVE [23] and STAR-SMIE [3, 46].

Dataset	Split Details (Unseen Category Indices)
SA-DAVE [23]:	
NTU-60 [38] (55/5 Split)	❶: [0, 8, 15, 28, 46] ❷: [15, 19, 23, 47, 50] ❸: [29, 37, 38, 45, 55]
NTU-120 [27] (110/10 Split)	❶: [0, 4, 6, 7, 24, 37, 54, 59, 97, 113] ❷: [63, 79, 86, 92, 98, 100, 103, 110, 111, 117] ❸: [9, 14, 17, 44, 60, 75, 81, 89, 108, 110]
PKU-MMD I [7] (46/5 Split)	❶: [10, 19, 27, 38, 48] ❷: [0, 9, 17, 30, 42] ❸: [18, 24, 31, 43, 45]
STAR-SMIE [3, 46]:	
NTU-60 [38] (55/5 Split)	❶: [4, 19, 31, 47, 51] ❷: [12, 29, 32, 44, 59] ❸: [7, 20, 28, 39, 58]
NTU-120 [27] (110/10 Split)	❶: [3, 18, 26, 38, 41, 60, 87, 99, 102, 110] ❷: [5, 12, 14, 15, 17, 42, 67, 82, 100, 119] ❸: [6, 20, 27, 33, 42, 55, 71, 97, 104, 118]
PKU-MMD I [7] (46/5 Split)	❶: [3, 14, 29, 31, 49] ❷: [2, 15, 39, 41, 43] ❸: [4, 12, 16, 22, 36]

C. Implementation Details

Following prior works [3, 11], we adopt Shift-GCN [6] as the skeleton encoder. For the text encoder, we use CLIP ViT-L/14@336px [34], consistent with [3, 4]. Both the encoder and decoder of the VAE are implemented as two-layer MLPs. For flow matching, we employ a single-layer DiT backbone [32]. The training consists of two stages: the "learning" phase and the "deciding" phase, with 1,000 and 200 iterations, respectively. We use the AdamW optimizer with a weight decay of 0.01 and a learning rate of 1×10^{-4}. Logit-normal sampling [9] is applied to bias the training timesteps in flow matching. The batch size is set to 64. The hyperparameters λ_{Align} and λ_{Flow} are both set to 0.1, and the GZSL threshold γ is fixed to 0.75. All experiments are implemented in PyTorch and conducted on a GeForce RTX 4090 Ti GPU. All ablation studies and qualitative analyses are used SynSE-based Shift-GCN features.

D. Additional Performance Comparison

Basic Split Benchmark Evaluation II. We further compare our method with previous approaches under the cross-view and cross-setup evaluation protocols. Since SynSE does not provide pre-trained skeleton features for these settings, we employ the STAR-based 1s-Shift-GCN skeleton features for a fair performance comparison. As shown in Table 10 and Table 11, our method consistently outperforms prior works on both ZSL and GZSL metrics, demonstrating its strong robustness to variations in camera view and setup conditions.

Random Split Benchmark Evaluation II. We also compare our method with other approaches under the random split strategies proposed in STAR-SMIE [3, 46], as shown in Table 12. The results demonstrate that our method remains robust across different split strategies and consistently outperforms prior works. Notably, our method even surpasses the two-stream approaches, such as Neuron [4], despite being a single-stream model without result stacking.

More Challenging Seen-Unseen Evaluation. We further evaluate the efficiency of our method under reduced seen category priors, as shown in Table 13. Even with fewer seen categories, our method still achieves competitive results compared with previous approaches, demonstrating its

Table 10. Performance comparisons on the Xview task of NTU-60 and Xset task of NTU-120. The best and the second-best results are marked in **Red** and Blue, respectively. All methods use STAR-based [3] Shift-GCN skeleton features, as SynSE [11] does not provide Xview and Xset features. ‡ denotes the two-stream fusion, while others are single-stream.

Method	Venue	NTU-60 (Xview)								NTU-120 (Xset)							
		55/5 Split				48/12 Split				110/10 Split				96/24 Split			
		ZSL	GZSL			ZSL	GZSL			ZSL	GZSL			ZSL	GZSL		
		$\mathcal{Acc}$	$\mathcal{S}$	$\mathcal{U}$	$\mathcal{H}$	$\mathcal{Acc}$	$\mathcal{S}$	$\mathcal{U}$	$\mathcal{H}$	$\mathcal{Acc}$	$\mathcal{S}$	$\mathcal{U}$	$\mathcal{H}$	$\mathcal{Acc}$	$\mathcal{S}$	$\mathcal{U}$	$\mathcal{H}$
ReViSE [18]	ICCV 2017	54.4	25.8	29.3	27.4	17.2	34.2	16.4	22.1	30.2	4.0	23.7	6.8	13.5	2.6	3.4	2.9
JPoSE [41]	ICCV 2019	72.0	61.1	59.5	60.3	28.9	29.0	14.7	19.5	52.8	23.6	4.4	7.4	38.5	79.3	2.6	4.9
CADA-VAE [37]	CVPR 2019	75.1	65.7	56.1	60.5	32.9	49.7	25.9	34.0	52.5	46.0	44.5	45.2	38.7	47.6	26.8	34.3
SynSE [11]	ICIP 2021	68.0	65.5	45.6	53.8	29.9	61.3	24.6	35.1	59.3	58.9	49.2	53.6	41.4	46.8	31.8	37.9
SMIE [46]	ACMMM 2023	79.0	-	-	-	41.0	-	-	-	57.0	-	-	-	42.3	-	-	-
STAR [3]	ACMMM 2024	81.6	71.9	70.3	71.1	42.5	66.2	37.5	47.9	65.3	59.3	59.5	59.4	44.1	53.7	34.1	41.7
STAR++ [5]	TCSVT 2026	81.9	61.6	71.5	66.2	50.6	60.8	41.1	49.0	69.0	63.4	49.6	55.7	50.4	57.7	39.3	46.8
Neuron‡ [4]	CVPR 2025	**87.8**	70.6	75.9	73.2	63.3	65.3	**58.1**	**61.5**	71.1	**67.5**	58.9	62.9	54.0	**67.0**	44.9	53.8
Flora (Ours)	This work	85.2	**82.7**	**76.5**	**79.5**	**64.9**	**75.4**	50.0	60.1	**76.0**	62.8	**65.1**	**63.9**	**63.7**	55.4	**56.3**	**55.9**

Table 11. Performance comparisons on PKU-MMD I dataset under the ZSL and GZSL setting. The best and the second-best results are marked in **Red** and Blue, respectively. All methods use STAR-based [3] Shift-GCN skeleton features, as SynSE [11] does not provide PKU-MMD features.

Method	Venue	PKU-MMD I (Xsub)								PKU-MMD I (Xview)							
		46/5 Split				39/12 Split				46/5 Split				39/12 Split			
		ZSL	GZSL			ZSL	GZSL			ZSL	GZSL			ZSL	GZSL		
		$\mathcal{Acc}$	$\mathcal{S}$	$\mathcal{U}$	$\mathcal{H}$	$\mathcal{Acc}$	$\mathcal{S}$	$\mathcal{U}$	$\mathcal{H}$	$\mathcal{Acc}$	$\mathcal{S}$	$\mathcal{U}$	$\mathcal{H}$	$\mathcal{Acc}$	$\mathcal{S}$	$\mathcal{U}$	$\mathcal{H}$
ReViSE [18]	ICCV 2017	54.2	44.9	34.5	39.1	19.3	35.7	13.0	19.0	54.1	50.7	39.9	44.6	12.7	34.5	9.4	14.8
JPoSE [41]	ICCV 2019	57.4	67.0	43.0	52.4	27.0	64.8	26.5	37.6	53.1	72.9	42.5	53.7	22.8	57.6	20.2	29.9
CADA-VAE [37]	CVPR 2019	73.9	76.2	51.8	61.7	33.7	69.0	29.3	41.1	74.5	79.9	61.5	69.5	29.5	62.4	28.3	39.0
SynSE [11]	ICIP 2021	69.5	**77.8**	40.2	53.0	36.5	71.9	30.0	42.3	71.7	69.9	51.1	59.0	25.4	61.9	22.6	33.1
SMIE [46]	ACMMM 2023	72.9	-	-	-	44.2	-	-	-	71.6	-	-	-	40.7	-	-	-
STAR [3]	ACMMM 2024	76.3	59.1	72.3	65.0	50.2	72.7	44.7	55.4	75.4	73.5	**72.2**	72.8	50.5	69.8	47.5	56.5
STAR++ [5]	TCSVT 2026	77.1	69.9	**73.5**	71.7	55.4	71.2	52.3	60.3	**76.6**	72.2	69.0	70.6	57.0	75.1	51.3	60.9
Flora (Ours)	This work	**79.1**	**76.0**	65.9	**70.6**	**55.4**	**74.5**	**52.3**	**61.5**	76.3	**76.0**	71.4	**73.7**	**58.7**	**77.2**	**55.6**	**64.6**

Table 12. Average performance comparison of three random seen-unseen splits on NTU-60 and PKU-MMD I datasets proposed by SMIE-STAR [3, 46] with Shift-GCN features. The best and the second-best results are marked in **Red** and Blue, respectively. ‡ denotes the two-stream fusion, while others are single-stream.

Method	NTU-60 55/5 (Xsub)		PKU-MMD I 46/5 (Xsub)	
	ZSL	GZSL	ZSL	GZSL
ReViSE [18]	54.7	27.4	48.7	32.8
JPoSE [41]	56.6	44.7	39.2	31.7
CADA-VAE [37]	58.0	47.1	49.0	52.7
SynSE [11]	59.9	49.9	43.5	40.4
SMIE [46]	64.2	-	66.4	-
STAR [3]	77.5	62.8	70.6	67.1
STAR++ [5]	79.5	62.4	73.6	68.3
Neuron‡ [4]	84.5	71.2	74.4	**69.2**
Flora (Ours)	**85.1**	**71.7**	**76.5**	68.4

superior generalization capability. Notably, under the 30/30 split setting on NTU-60 (Xsub), our method shows a substantial improvement, highlighting its strong potential when trained with limited seen category priors.

Table 13. Performance comparisons on NTU-60 and NTU-120 with more challenging seen-unseen splits. The best and the second-best results are marked in **Red** and Blue, respectively. All methods use our own pre-trained Shift-GCN skeleton features, as PURLS [47] and TDSM [8] do not provide their pre-trained models.

Method	Venue	NTU-60 (Xsub)		NTU-120 (Xsub)	
		40/20	30/30	80/40	60/60
ReViSE [18]	ICCV 2017	24.3	14.8	19.5	8.3
JPoSE [41]	ICCV 2019	20.1	12.4	13.7	7.7
CADA-VAE [37]	CVPR 2019	16.2	11.5	10.6	5.7
SynSE [11]	ICIP 2021	19.9	12.0	13.6	7.7
PURLS [47]	CVPR 2024	31.1	23.5	28.4	19.6
ScoPLe [48]	CVPR 2025	32.0	18.2	25.3	15.7
TDSM [8]	ICCV 2025	**36.1**	25.9	37.0	27.2
Flora (Ours)	This work	31.1	**35.7**	**40.1**	**29.0**

Per-instance Inference Time Comparison. In Table 15, we report the inference time as the number of candidate categories increases during per-instance inference. Notably, our method maintains an inference time of under one second even when matching against 1000 categories.

Table 14. ZSL Comparison with other methods under low-shot training with SynSE-based [11] Shift-GCN features.

Method	NTU-60								NTU-120							
	55/5 (Xsub)				48/12 (Xsub)				110/10 (Xsub)				96/24 (Xsub)			
	1%	5%	10%	50%	1%	5%	10%	50%	1%	5%	10%	50%	1%	5%	10%	50%
ReViSE [18]	51.0	58.0	58.0	56.9	9.8	11.1	15.6	15.6	14.8	14.8	23.6	20.3	5.2	7.3	7.8	8.5
JPoSE [41]	33.0	46.8	62.1	65.0	23.8	25.8	28.3	32.6	15.6	36.5	49.9	48.2	8.6	33.5	33.9	35.7
CADA-VAE [37]	76.6	74.6	76.9	74.1	24.3	26.4	27.6	26.8	29.9	38.3	39.1	35.3	25.4	26.1	25.0	25.4
SynSE [11]	44.3	43.7	42.8	43.8	18.6	17.1	17.3	18.4	56.0	55.7	56.0	56.2	24.1	26.5	26.1	25.5
SMIE [46]	43.8	77.1	76.9	77.6	29.3	36.0	38.1	40.2	36.0	55.9	58.1	60.8	13.9	30.4	34.4	42.7
SA-DAVE [23]	21.1	45.1	60.4	81.3	18.7	16.5	20.0	30.4	14.4	28.5	40.9	55.6	9.5	12.3	21.9	34.7
STAR [3]	40.6	75.5	77.0	79.1	11.6	32.9	35.3	37.5	18.9	41.8	46.8	53.2	8.6	31.3	33.0	34.5
Neuron [4]	47.7	76.9	79.4	81.5	20.7	34.1	45.3	52.8	28.8	48.5	62.8	68.6	10.2	22.8	33.5	51.0
FS-VAE [42]	79.3	79.3	79.4	78.9	38.0	38.7	38.7	38.7	72.7	69.6	69.6	70.2	50.2	49.7	47.9	48.5
TDSM [8]	78.5	80.7	82.3	83.8	32.1	49.2	52.4	51.5	63.3	69.3	66.3	71.9	43.9	49.1	55.1	59.7
Flora (Ours)	**82.8**	**86.5**	**85.6**	**85.6**	**46.5**	**54.3**	**56.1**	**55.4**	**77.4**	**78.9**	**78.1**	**78.1**	**58.0**	**65.1**	**65.9**	**65.8**

Table 15. Per-instance Inference Time

# Cand. Classes	5	10	50	100	500	1000
Time (ms)	4.5	7.1	26.9	49.9	252.3	511.0

E. Additional Ablation Studies

Influence of Learning and Deciding Phases. In Table 16, we analyze the contributions of the learning and deciding phases within the overall framework. The neighbor-aware mechanism plays a crucial role, indicating that high-quality cross-modal alignment serves as a cornerstone for zero-shot skeleton-based action recognition. Furthermore, when equipped with the open-flow classifier, the framework better preserves information during the recognition stage, leading to improved performance.

Table 16. Component analysis on learning and deciding phases. †baseline alignment (Sec. 3). ‡similarity matching. §calibration strategy in [3, 4].

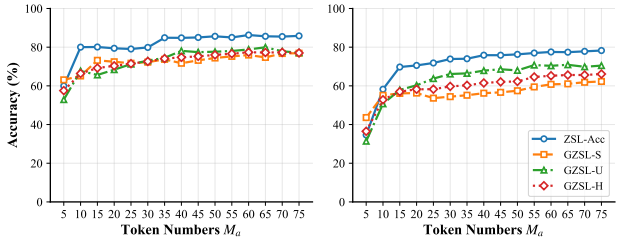
Neighbor-aware Semantic	Open-form Flow	NTU-60 (48/12)		NTU-120 (110/10)	
		ZSL	GZSL	ZSL	GZSL
✗†	✗‡	48.2	42.3§	71.1	55.8§
✗†	✓	49.6	46.3	74.8	63.9
✓	✗‡	56.7	51.6§	77.1	64.9§
✓	✓	**65.3**	**60.5**	**79.6**	**66.1**

Influence of Text Encoders. As shown in Table 17, the performance varies across different text encoders. Despite these discrepancies, the overall results remain strong under the ZSL setting. Interestingly, the best performance is not achieved with the most powerful model, *i.e.*, ViT-H/14. For fairness and consistency with prior studies [3, 4], we adopt the ViT-L/14@336px model in all experiments.

Influence of Token Numbers M_a. As shown in Fig. 8, the performance of our method improves substantially as the number of tokens increases, and it gradually converges to a stable level when more tokens are involved. This trend suggests that enriching semantic representations contributes to more effective cross-modal alignment and that a sufficient number of tokens is essential to fully capture the semantic diversity required for robust performance.

Table 17. Analysis of different text encoders on NTU-120 (Xsub).

Text Encoder	110/10 Split		96/24 Split	
	ZSL	GZSL	ZSL	GZSL
ViT-B/32	77.1	61.2	62.1	47.9
ViT-B/16	77.7	63.9	62.5	46.4
ViT-L/14	79.6	66.3	65.6	52.6
ViT-L/14@336px	79.6	66.1	66.4	53.2
ViT-H/14	73.5	62.6	66.3	52.0

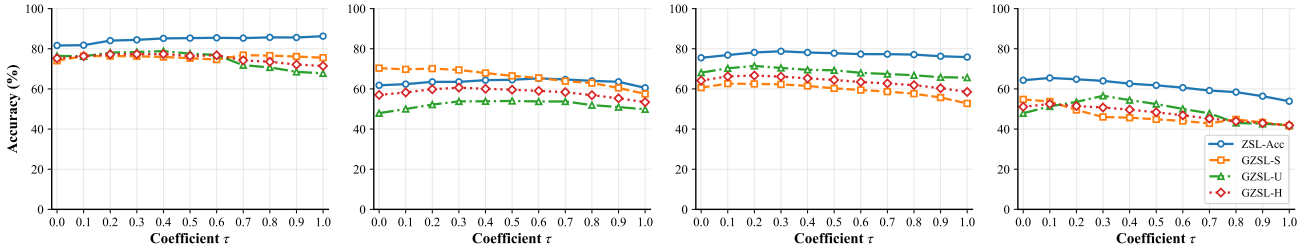


(a) NTU-60 (55/5) (b) NTU-120 (110/10)

Figure 8. Performance comparison on NTU-60 and NTU-120 under varying token numbers M_a.

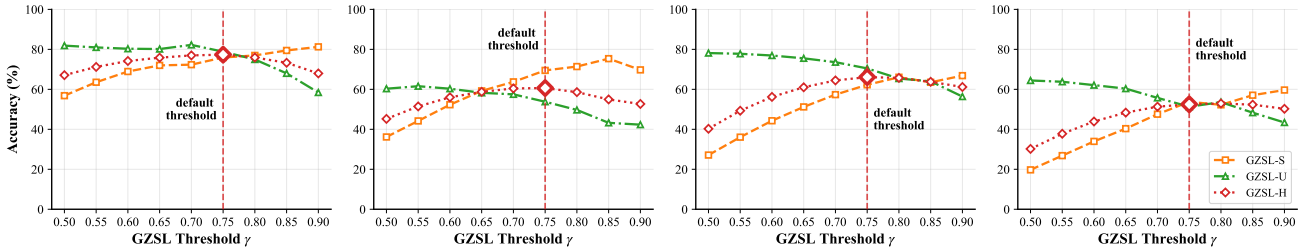
Influence of Coefficient τ. As illustrated in Fig. 9, both the harmonic accuracy and unseen performance first increase for smaller values of τ and then drop as τ varies. Overall, the performance trend stabilizes at a relatively high level, indicating that τ serves as a trade-off parameter that balances inter-class discriminability and the smoothness of the semantic space. Additionally, this coefficient is also robust to the selection of values.

Influence of Threshold γ **in GZSL Prediction**. As shown in Fig. 10, the GZSL performance is sensitive to the threshold γ, which controls whether a skeleton sample is classified as belonging to the seen or unseen domain. This behavior is expected, as γ directly governs the gating mechanism in domain prediction. A higher threshold biases the model to-



(a) NTU-60 (55/5) (b) NTU-60 (48/12) (c) NTU-120 (110/10) (d) NTU-120 (96/24)

Figure 9. Performance comparison on NTU-60 and NTU-120 under varying coefficient τ.



(a) NTU-60 (55/5) (b) NTU-60 (48/12) (c) NTU-120 (110/10) (d) NTU-120 (96/24)

Figure 10. GZSL Performance comparison on NTU-60 and NTU-120 with varying predefined threshold γ.

ward assigning skeleton samples to the seen domain, while a lower value favors the unseen domain.

Influence of Distribution Alignment Coefficient $\lambda_{\mathbf{Align}}$ in Learning Phase. As shown in Fig. 11, the performance exhibits an overall trend of increasing initially and then decreasing. When $\lambda_{\mathbf{Align}}$ exceeds 0.1, the performance drops sharply, particularly on the seen domains. This suggests that large $\lambda_{\mathbf{Align}}$ may cause the latent space to collapse, weakening the dominance of the reconstruction objectives.

Influence of Contrastive Regularization Coefficient λ_{Flow} in the Deciding Phase. As illustrated in Fig. 12, the performance remains stable for smaller values of λ_{Flow} but declines as the coefficient increases. This suggests that mild contrastive regularization is beneficial for enhancing generalization, whereas an overly strong contrastive objective may hinder the classifier's discriminative capability, especially for the seen categories.

Influence of Timestep Sampling Types in the Deciding Phase. As shown in Table 18, we compare the uniform-based and logit-based timestep sampling strategies. The results indicate that the choice of sampling type has minimal impact on the training of the flow classifier. In this work, we adopt the logit-based sampling strategy for consistency.

Influence of Flow Matching Backbone. As shown in Table 19, we further investigate the performance of flow classifiers with different backbone architectures. Even with a simple two-layer MLP, the ZSL performance remains strong, showing only a slight degradation compared to a single-layer DiT block. This indicates that our flow classifier is largely independent of architectural complexity and can achieve competitive results with minimal network design.

Table 18. Analysis of timestep sampling types in the deciding phase.

Types	110/10 Split		96/24 Split	
	ZSL	GZSL	ZSL	GZSL
Uniform-based	78.6	65.6	65.7	52.0
Logit-based (Ours)	79.6	66.1	66.4	53.2

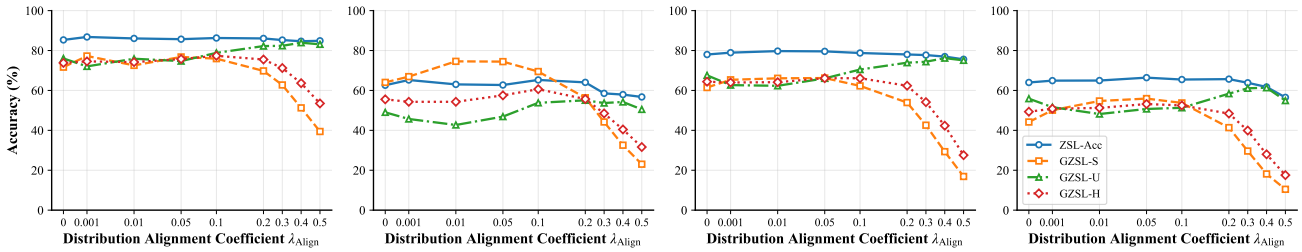
Table 19. Analysis of flow matching backbone in deciding phase.

Direction	110/10 Split		96/24 Split	
	ZSL	GZSL	ZSL	GZSL
MLP	78.6	65.3	65.1	51.4
DiT (Ours)	79.6	66.1	66.4	53.2

Influence of Flow Directions in the Deciding Phase. As shown in Table 20, the choice of flow direction between skeleton and semantics has little effect on performance, since flow matching operates on interpolated vectors between the two modalities. In this work, we set the default flow direction from semantics to skeleton.

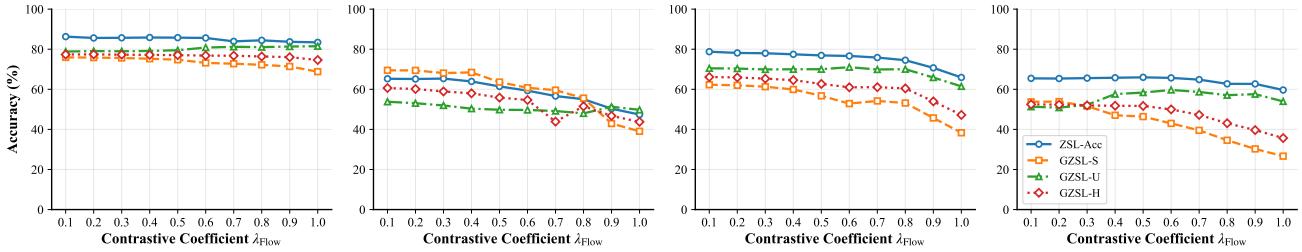
F. Additional Discussions

Skeleton Perspective. We summarize and discuss zero-shot skeleton recognition from the perspective of skeleton



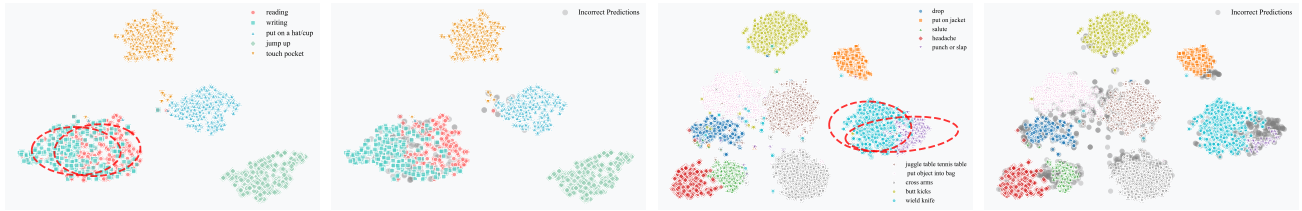
(a) NTU-60 (55/5) (b) NTU-60 (48/12) (c) NTU-120 (110/10) (d) NTU-120 (96/24)

Figure 11. Performance comparison on NTU-60 and NTU-120 under various distribution alignment coefficient λ_{Align} in the learning phase.



(a) NTU-60 (55/5) (b) NTU-60 (48/12) (c) NTU-120 (110/10) (d) NTU-120 (96/24)

Figure 12. Performance comparison on NTU-60 and NTU-120 under different contrastive regularization coefficient λ_{Flow} in the deciding phase.



(a) NTU-60 (55/5) (b) NTU-60 (55/5) (c) NTU-120 (110/10) (d) NTU-120 (110/10)

Figure 13. t-SNE visualization on NTU-60 (55/5) and NTU-120 (110/10).

Table 20. Analysis of flow directions on the NTU-120 (Xsub) in the deciding phase.

Direction	110/10 Split		96/24 Split	
	ZSL	GZSL	ZSL	GZSL
Skeleton$\mathcal{N}_s$ ⇒Semantic$\mathcal{N}_a$	78.9	65.8	65.3	52.2
Semantic$\mathcal{N}_a$ ⇒Skeleton$\mathcal{N}_s$ (Ours)	79.6	66.1	66.4	53.2

as follows:

- *Low-shot Training Samples*. As shown in Table 14, our experiments demonstrate the promising potential of zero-shot skeleton action recognition toward more efficient learning. This observation motivates us not only to focus on limited seen categories but also to explore learning from limited samples. Such a direction suggests that it is feasible to build an intelligent system with strong generalizability and robustness, even when trained with a small number of samples from a few categories.
- *Representation Quality*. Another key challenge lies in the limited information contained in skeleton data. For instance, a single joint is often used to represent an entire hand, which leads to overlapping skeleton features across similar actions (Fig. 13), such as reading and writing. This overlap makes it difficult to separate features from different categories, particularly for unseen ones, since their priors are unavailable during training. Therefore, incorporating finer-grained skeleton representations—such as increasing the number of joints to capture more detailed motion—may be a promising direction for advancing skeleton-based community, beyond the zero-shot setting.

Semantic Perspective. We further discuss it from the perspective of semantics as follows:

- *Skeleton-specific Semantics*. Current semantics are typically action-specific, whether derived from hand-crafted labels or LLM-generated descriptions, and thus are not inherently aligned with the nature of skeleton representations. For instance, the semantics of "pick up" share

little linguistic similarity with "put on a shoe", yet their skeleton sequences are highly similar, as both involve a squatting motion. This discrepancy, where distant semantics correspond to highly similar skeletal patterns, leads to cross-modal structural inconsistency prior to alignment. On such a fragile foundation, building a reliable semantic–skeleton alignment becomes inherently difficult. Therefore, designing skeleton-structural semantics that are consistent with the physical motion patterns is crucial, though largely overlooked in existing research.

- *Semantic Diversity*. Action descriptions can vary significantly across observation viewpoints or subjects with different body shapes. Incorporating diverse semantics that account for these variations is essential for achieving robust alignment. A promising direction is to leverage sample-level semantics for alignment. It effectively reframes the recognition task as a zero-shot captioning problem, where the model learns to describe actions through semantically grounded understanding rather than rigid label matching. In this setting, we believe **`Flora`** can play a vital role.

Algorithm Perspective.

- *Alignment*. Similar to how a limited set of pixels with diverse combinations can generate an infinite number of images and promote zero-shot learning in various domains, exploring the compositionality of skeletal primitives (such as fixed joint groups or joint motion velocities) is equally important. A finite number of primitives with different variations can represent an unlimited range of actions. In contrast to existing paradigms that rely solely on pre-extracted skeleton features for alignment, developing a continual skeleton composition framework can enable cross-modal alignment at a more fundamental level, thereby enhancing the performance of zero-shot skeleton-related tasks.
- *Task*. Beyond recognition, building a skeleton-based foundation model capable of handling various tasks, including skeleton captioning and generation, under zero-shot settings is a promising direction. Our proposed **`Flora`** framework provides a paradigm for these advancements by establishing a dynamic flow-based pathway between skeletons and semantics, effectively bridging the gap between perception and understanding.