Supplementary Material
MMCM: Multimodality-aware Metric using Clustering-based Modes for Probabilistic Human Motion Prediction

Kyotaro Tokoro, Hiromu Taketsugu, Norimichi Ukita
Toyota Technological Institute
{sd24439,sd25502,ukita}@toyota-ti.ac.jp

A. Introduction

We show the following additional results and analyses:

1. Validity of our hyper parameter setting (Sec. B).
2. Robustness of MMCM's hyper parameters (Sec. C).
3. Validity of our clustering method choice (Sec. D).
4. Validity of our mode definition (Sec. E).
5. Additional qualitative results (Sec. F).
6. Additional other metrics results (Sec. G).
7. Explicit algrorithm (Sec. H).
8. Detailed description of the threshold τ (Sec. I).
9. Consideration for kinematic constraints(Sec. J).
10. MMCM for 2D trajectory prediction (Sec. K)

B. Hyper Parameter Setting

This section explains how we chose the hyper parameters of the dimensionality reduction and HDBSCAN [15] to show that our hyper parameter settings are reasonable.

The dimensionality reduction has two hyper parameters: "AE_dim", which is the latent dimensionality after the autoencoder, and "UMAP_dim", which is the dimensionality after UMAP [16]. HDBSCAN also has two hyper parameters: "min_cluster_size", which sets the minimum cluster size, and "min_samples", which controls the sensitivity to noise. Because MMCM depends on these values, they must be selected carefully.

We select hyper parameters based on the clustering results, because the objective of employing the autoencoder, UMAP, and HDBSCAN is to achieve stable clustering. To measure stable clustering, we use the criteria, which jointly consider the noise rate and the sum of cluster persistence. The noise rate is the percentage of data points that are not assigned to any cluster. A high noise rate indicates that small but certain clusters are dismissed. The cluster persistence is the lifetime of a cluster in the HDBSCAN condensed tree. A high cluster persistence indicates that the cluster is more stable. A lower noise rate is preferable. So, we employ HDBSCAN's approximate-prediction feature to assign clusters to the test data and thus detect the noise rate for the test set. However, tuning HDBSCAN to obtain a noise rate of zero forces sparse regions to be treated as dense, producing unnaturally large clusters. This tuning result is not suitable. We therefore require the noise rate to be no greater than one-third, which prevents an excessive number of points from being classified as noise while avoiding over-suppressing noise. Among hyper parameter settings satisfy the noise rate criterion, we select setting whose cluster persistence is maximam. The hyper parameters selected by the criterion can avoid both over-segmentation and under-segmentation. Here, we call this criterion the noise–persistence criterion.

Table 2. Hyper parameter search range.

Parameters	Range
AE_dim	{16, 32, 64, 128, 256}
UMAP_dim	{2, 4, 8, 16}
min_cluster_size	{2, 3, 4, 5, 7, 10, 15, 20, 25, 30, 40, 50, 60}
min_samples	{1, 2, 3, 4, 5, 7, 10, 15, 20, 25, 30, 40, 50, 60}

We performed a search over the ranges listed in Table 2. The search results for each dataset are reported below.

Human3.6M Table 2 lists 3,640 candidate hyper parameter combinations. Presenting all results would be difficult to understand, so Table 3 reports representative results.

We first examine the effect of the autoencoder dimension (AE_dim). The UMAP_dim is fixed at 2, and AE_dim is varied across several values. For each choice of AE_dim, we select the HDBSCAN parameters that meet the noise–persistence criterion. The results correspond to Ref. idx = {1,2,3,4,5}. The cluster persistence increases from 16 to 64 latent dimensions but decreases beyond 64, indicating that an AE dimension of 64 is optimal. Next, we examine the effect of the UMAP dimension (UMAP_dim). We fix AE_dim at 64 and sweep the UMAP dimension,

Table 3. Results of hyper parameter search in H36M. Red and blue indicate the best and second-best results.

Ref. idx	AE_dim	UMAP_dim	min_cluster_size	min_samples	noise rate (%)	cluster persistence ↑
1	16	2	50	4	12	6.36
2	32	2	15	1	33	18.24
3	**64**	**2**	**15**	**1**	**32**	**21.93**
4	128	2	20	1	32	**21.06**
5	256	2	29	1	33	20.54
6	64	4	20	4	10	9.31
7	64	8	20	2	8	12.03
8	64	2	2	1	49	17.23
9	64	2	50	1	7	6.27
10	64	2	15	10	10	9.36

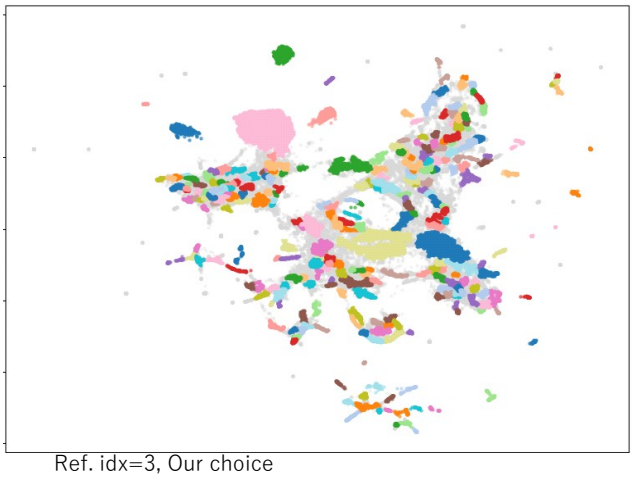
Table 4. Results of hyper parameter search in AMASS. Red and blue indicate the best and second-best results.

Ref. idx	AE_dim	UMAP_dim	min_cluster_size	min_samples	noise rate (%)	cluster persistence ↑
1	16	2	60	1	40	21.61
2	32	2	60	1	32	16.75
3	**64**	**2**	**50**	**1**	**31**	**22.43**
4	128	2	50	1	30	**21.76**
5	256	2	60	5	30	17.83
6	64	4	50	1	43	20.33
7	64	8	50	20	43	17.71
8	64	2	10	1	41	1.56
9	64	2	50	60	23	12.25

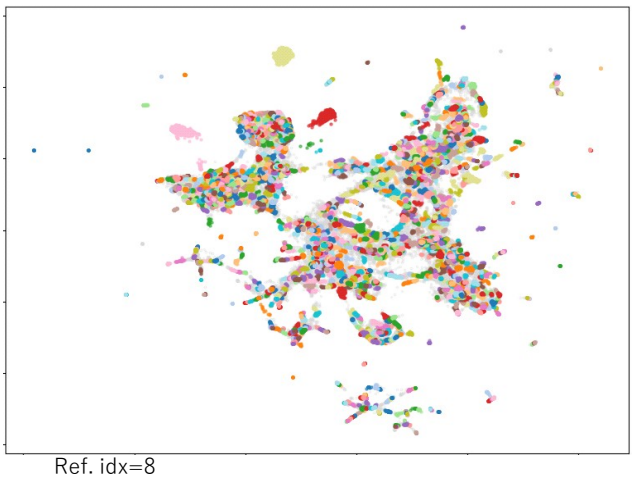
again selecting the best HDBSCAN parameters for the noise–persistence criterion. Ref. idx = {3,6,7} show that a two-dimensional UMAP embedding is the most stable. Finally, with AE_dim = 64 and UMAP_dim = 2, we vary HDBSCAN's min_cluster_size and min_samples. Results appear as Ref. idx = {3,8,9,10}. Some combinations mark nearly half the data as noise or yield very low cluster persistence. Figure 11 visualises the corresponding clustering results; grey points denote data classified as noise. From top to bottom, the plots correspond to Ref. idx = {3,8,9}. Although Ref. idx 3 seems to differ from Fig. 7 in the main paper, the two plots match after re-assigning noise points to their nearest cluster centroid as described in the main paper. Ref. idx 8 yields 4,254 clusters, which is clearly an over-segmentation, whereas Ref. idx 9 merges almost the entire distribution on the right-hand into a single cluster. In contrast, the setting adopted by MMCM (top) avoids both extreme results, demonstrating that the chosen hyper parameters have an appropriate balance.

AMASS We perform the same hyper parameter analysis on the AMASS dataset. Representative combinations

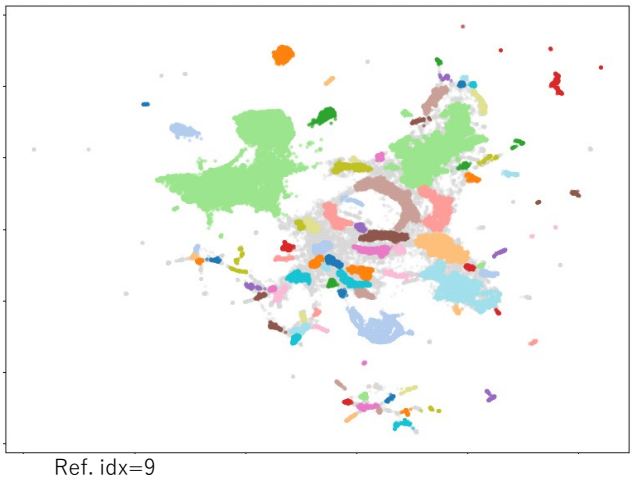
Figure 11. Clustering results with other parameters in H36M. Top: Ref. idx=3, Center: Ref. idx=8, Bottom: Ref. idx=9



Ref. idx=3, Our choice



Ref. idx=8



Ref. idx=9

Figure 12. Clustering results with other parameters in AMASS. Top: Ref. idx=3, Center: Ref. idx=8, Bottom: Ref. idx=9

are listed in Table 4. As with H36M, comparing Ref. idx = {1,2,3,4,5} shows that an AE_dim of 64 is appropriate, while the comparison of Ref. idx = {1,6,7} indicates that a UMAP_dim of 2 is optimal. Next, we examine the HDBSCAN parameters with AE_dim fixed at 64 and UMAP_dim fixed at 2; the relevant rows are Ref. idx = {3,8,9}. Some settings drive the cluster persistence to extremely low values. Figure 12 visualizes these clustering results. The middle plot (Ref. idx 8) yields 1,803 clusters, an obvious case of over-segmentation. The bottom plot (Ref. idx 9), in contrast, produces too few clusters and is likely to group more different motions in the same cluster. The top plot, corresponding to the hyper parameter combination adopted by MMCM, avoids both extremes, demonstrating that the chosen hyper parameters are well balanced.

Discussions As noted above, our hyperparameter selection criteria are clear, and the settings can be determined with a simple search. In practice, our official repository provides a script (./compute_mmcm/parameter_search.py) that grid searches these parameters (latent dimension, min_cluster_size, and min_samples) and tunes them automatically in about 20 minutes (training the autoencoder, if needed, takes a few additional hours).

Moreover, we argue that reasonable defaults can be set even without an explicit hyperparameter search. Comparing Human3.6M and AMASS, we find that AE_dim, UMAP_dim, and min_samples are identical (64, 2, and 1, respectively). This suggests that, for other datasets, these three hyperparameters are likely to take the same—or very similar—values. By contrast, min_cluster_size differs between Human3.6M and AMASS because it scales with dataset size: as it controls the minimum cluster size, smaller datasets require smaller values, whereas larger datasets require larger ones. In our experiments, Human3.6M (14,847 sequences) used min_cluster_size = 15, while AMASS (60,272 sequences) used 50, consistent with this trend. From these two points, we obtain a simple linear rule: for a new dataset of size x, set min_cluster_size $\approx x/1000$ to $x/1200$. These experimental findings will make hyperparameter setting easier.

C. Robustness of MMCM with Respect to Hyperparameters

In this section, we examine the metric's robustness to hyperparameter choices on AMASS under the same experimental setting as mentioned in Fig. 10 about H36M. The hyperparameters of MMCM include the embedding dimensionality and the HDBSCAN clustering parameters "min_cluster_size" and "min_samples". We conducted comprehensive experiments on all of them. Specifically, the embedding dimensionality was controlled by varying UMAP's

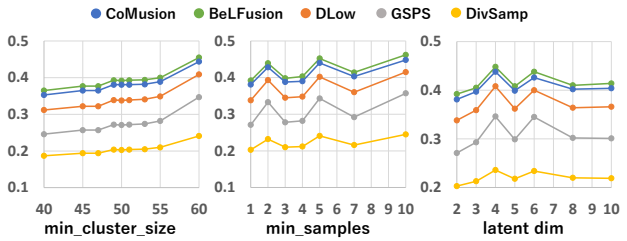


Figure 13. MMCM sensitivity to the some hyper parameters on AMASS. For all hyperparameters, changing our selected settings does not affect the relative ranking of HMP methods.

Table 5. Results of our two-stage pipeline and other dimensionality reduction methods in H36M. **Red** and **blue** indicate the best and second-best results.

Method	Dim	Noise rate (%)	Cluster persistence ↑	Number Cluster
No reduction	4,944	66	7.49	157
Autoencoder	64	97	8.62	26
Autoencoder	2	51	18.82	287
PCA	2	57	15.38	306
Autoencoder +PCA	2	48	15.35	321
UMAP	2	36	**20.80**	297
Autoencoder + UMAP (Ours)	**2**	**32**	**21.93**	**324**

output dimension. Figure 13 shows how the MMCM scores of the HMP methods (CoMusion [22], BeLFusion [4], DLow [24], GSPS [14], DivSamp [6]) change as these hyperparameters are varied.

Across all hyperparameters, changing their values slightly affects the absolute MMCM scores but not the relative ranking of methods. Since a key requirement for an evaluation metric is to discriminate between methods, these results indicate that MMCM is largely insensitive to hyperparameter choices; in other words, MMCM is robust to hyperparameter settings.

D. Comparison with Other Dimensionality Reduction Methods

We employ a two-stage dimensionality reduction pipeline that first encodes each motion with an autoencoder and then applies UMAP. To demonstrate that this choice is the most suitable preprocessing for clustering, we compare it with five alternatives:

1. No reduction,
2. Autoencoder,
3. PCA,
4. Autoencoder + PCA
5. UMAP,
6. Autoencoder + UMAP (Ours),

Table 6. Results of our two-stage pipeline and other dimensionality reduction methods in AMASS. **Red** and **blue** indicate the best and second-best results.

Method	Dim	Noise rate (%)	Cluster persistence ↑	Number Cluster
No reduction	7,749	97	8.14	29
Autoencoder	64	10	0.04	2
Autoencoder	2	20	8.22	254
PCA	2	51	5.93	242
Autoencoder +PCA	2	3	0.02	3
UMAP	2	28	**19.65**	239
Autoencoder + UMAP (Ours)	**2**	**31**	**22.43**	**278**

Since the purpose of the dimensionality reduction is to achieve stable clustering, we compare these dimensionality reduction methods by evaluating the clustering results. After applying each dimensionality reduction method to both datasets, we perform HDBSCAN and record the resulting cluster persistence, noise rate, and number of clusters.

The results in H36M are summarised in Table 5. Our "Autoencoder + UMAP" pipeline achieves the highest cluster persistence while keeping the noise rate low. The results in AMASS are summarised in Table 6. As same as results in H36M, "Autoencoder + UMAP" pipeline achieves the highest cluster persistence. These results in both datasets confirm that our pipeline is the best option for dimensionality reduction before clustering.

E. Mode Definition

This section contains two subsections. Section E.1 demonstrates the superiority of our choice of clustering approach by comparing it with several alternative clustering approaches. Section E.2 shows action label distribution and subject label distribution in human motion space of H36M and each dataset distribution in human motion space of AMASS.

E.1. Comparison with Other Clustering Methods

We adopt HDBSCAN as our clustering approach. In this section, we demonstrate that HDBSCAN is well-suited for clustering human motion space by comparing it with four alternatives: k-means [12], k-means++ [2], BIRCH [26], and Agglomerative Clustering [17]. Although two-dimensional visualizations, such as Fig. 14, offer intuition, they do not allow for a quantitative comparison be-

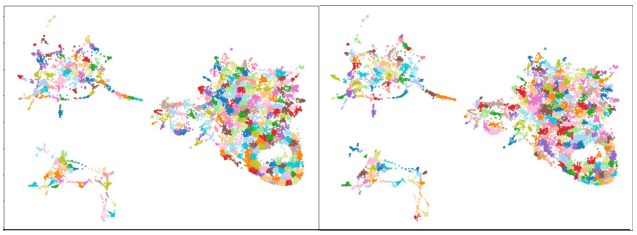


Figure 14. Visualizations of the two-dimensional motion space clustered by k-means++ or HDBSCAN.

Table 7. Comparison of HDBSCAN and other clustering approaches. Red and blue indicate the best and second-best results for each metric.

Clustering method	Sil ↑	DB ↓	CH ↑
k-means	0.419	0.744	53248
k-means++	**0.436**	**0.714**	**61332**
BIRCH	0.380	0.807	39806
Agglomerative Clustering	0.412	0.738	**55604**
HDBSCAN (Ours)	**0.461**	**0.636**	**4**9241

tween different clustering approaches. We therefore evaluate all methods with three standard quantitative metrics:

- Silhouette Coefficient (Sil) [20] – the mean difference between a sample's intra-cluster distance and its nearest-cluster distance; higher values indicate better separation.
- Davies–Bouldin Index (DB) [7] – the average similarity between each cluster and its most similar cluster; lower values indicate better separation.
- Calinski–Harabasz Index (CH) [5] – the ratio of between-cluster dispersion to within-cluster dispersion; higher values indicate more distinct clusters.

Table 7 reports the scores for all five approaches. Here, since clustering methods other than HDBSCAN require the number of clusters in advance, we set this number to the same number of clusters obtained by HDBSCAN. HDBSCAN achieves the best Silhouette Coefficient (Sil) and the lowest Davies–Bouldin Index (DB), marking it as the top performer on two of the three criteria. k-means++ records the highest Calinski–Harabasz Index (CH), but its Sil and DB scores are poorer than HDBSCAN. Taken together, these results confirm that HDBSCAN is the most suitable clustering approach for partitioning the human motion space.

E.2. Human Motion Space

Human3.6M. Because the H36M dataset provides both action and subject labels, we visualised the human motion space by coloring the points with these labels. The results

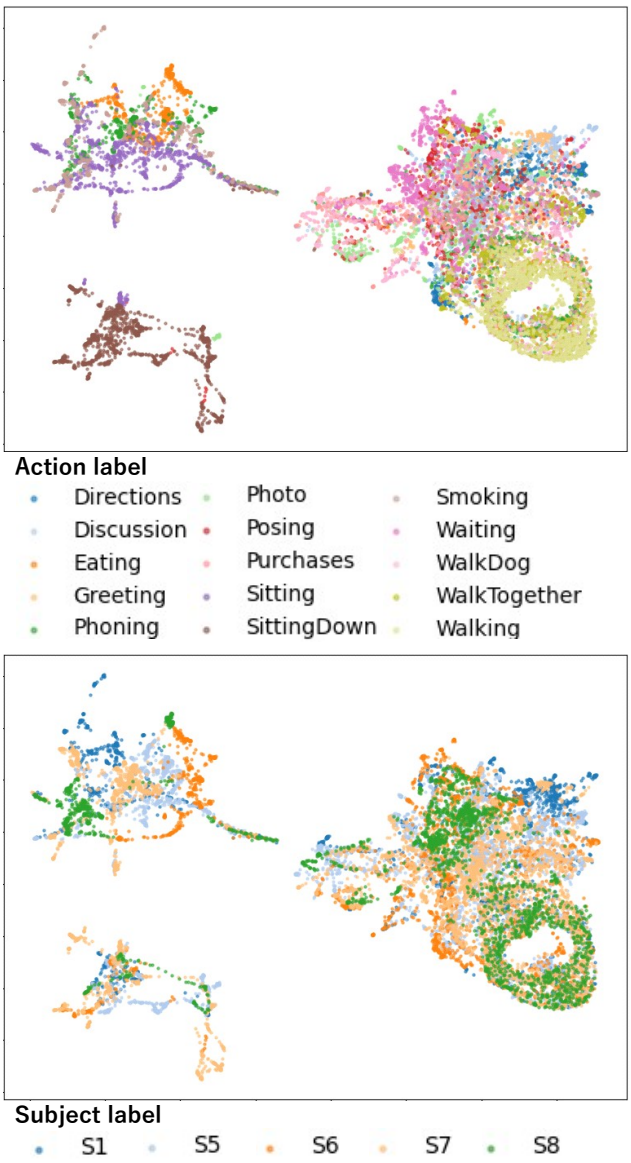


Figure 15. Two-dimensional motion space of H36M after Autoencoder + UMAP reduction. Top: points are colored by action label. Bottom: the same points, colored instead by subject label.

are shown in Fig. 15. Note that an action label is labeled to the entire source video, not to each sequence extracted by the sliding window. For example, a "Smoking" sequence may contain segments of walking or sitting.

In the action label plot, action labels such as "WalkTogether" and "Walking" appear close to each other because they are kinematically similar. Moreover, they are clearly separated from most other actions. In contrast, a broad label that contains various motions, such as "Posing", spreads across much of the space. A similar pattern emerges in the subject label plot. Data from the same subject form local

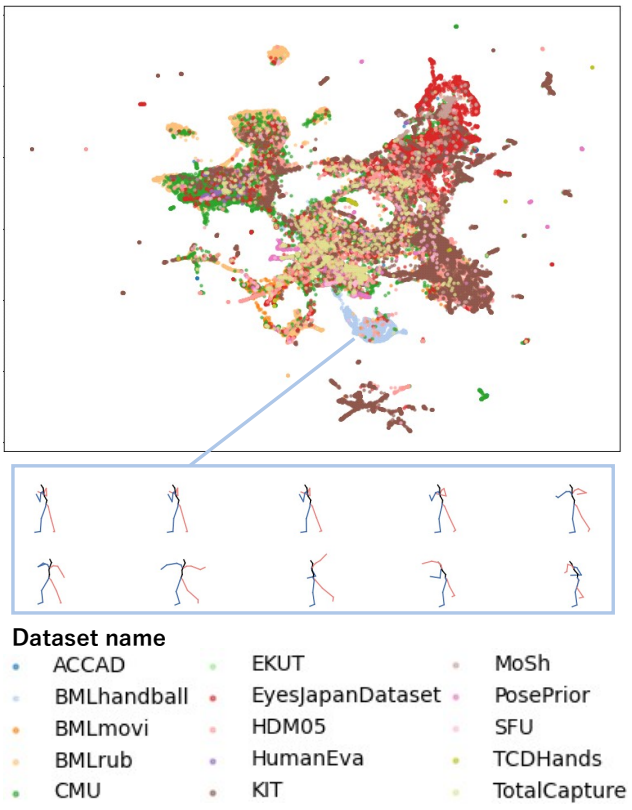


Figure 16. Two-dimensional motion space of AMASS colored by each dataset in AMASS.

groups, but each subject performs multiple actions, so their sequences still scatter over the space.

These observations show that neither action labels nor subject labels cleanly partition the human motion space. Therefore, defining modes by clustering, not using labels, is the appropriate choice for our method.

AMASS. Because the AMASS dataset is composed of multiple independent datasets, we visualised the human motion space by coloring the points with these datasets. The results are shown in Fig. 16.

Some datasets cluster tightly in latent space. For example, motions from the BMLhandball dataset concentrate in a small region because motions feature the distinctive "throwing a ball" pose shown beneath Fig. 16. By contrast, motions from many other datasets are widely scattered. These observations confirm that dataset names do not provide a reliable criterion for mode definition.

F. Additional Qualitative Results

This section shows additional results for Fig. 4 (Sec. F.1) and Fig. 5 (Sec. F.2) in the main paper. These results indicate that MMCM meets two requirements for multimodality: (a) coverage and (b) validity.

Table 8. Quantitative comparison with other metrics. Red and blue indicate the best and second-best results for each metric.

Method	Human3.6M				AMASS		
	MMCM↑	NLL↓	CMD↓	FID↓	**MMCM↑**	NLL↓	CMD↓
TPK	0.520	-5.35	6.326	0.538	0.374	-3.47	17.127
DLow	0.487	-3.09	4.927	1.255	0.314	-3.33	15.185
GSPS	0.417	-2.94	10.758	2.103	0.256	-2.83	18.404
DivSamp	0.409	-3.69	11.692	2.083	0.198	-3.01	50.239
HumanMAC	0.504	-5.35	-	-	-	-	-
BeLFusion	0.509	-2.70	5.988	0.209	0.386	-3.17	16.995
CoMusion	0.521	-4.94	3.202	0.102	0.378	-5.27	9.636

F.1. Additional Results for (a) Coverage

This section presents additional results that complement Fig. 4 in the main paper. In Fig. 17, the top row visualizes all 50 predicted motions, whereas the bottom row shows one representative motion from each mode. Across all sequences, the qualitative impression of multimodality is consistent with the quantitative MMCM scores.

F.2. Additional Results for (b) Validity

This section presents additional results of Fig. 5 in the main paper. Outlier motions excluded from valid modes are shown in Fig. 18. In every sequence, we observe motions with unrealistically elongated limbs or otherwise kinematically impossible poses. MMCM correctly flags these motions as abnormal, demonstrating that it meets (b) validity.

G. Comparison with Realism Metrics and Probabilistic Metrics

In Table 1 of the main manuscript, we compare multimodality metric (MMCM) with diversity metrics and accuracy metrics. In this section, we compare MMCM with additional metrics, which are realism metrics (FID and CMD) and probabilistic metrics (Negative Log-Likelihood, NLL). Realism metrics are described in Sec. 2.2 of the main paper. NLL evaluates the predictive distribution in probabilistic approaches and is commonly used in the 2D trajectory prediction task. To enable comparison across a broad set of motion prediction methods, following Salzmann et al. [21], we approximate NLL using Kernel Density Estimate-based Negative Log-Likelihood (KDE-NLL). Concretely, KDE-NLL is the mean negative log-likelihood of the GT under a distribution obtained by fitting a kernel density estimate to the predicted samples. These results are shown in Table 9. Here, computing FID requires class labels; since AMASS has not class lavels, we report FID only on Human3.6M.

We observe that MMCM and NLL do not correspond. This is because they evaluate different objectives, although both are used to evaluate probabilistic methods. NLL

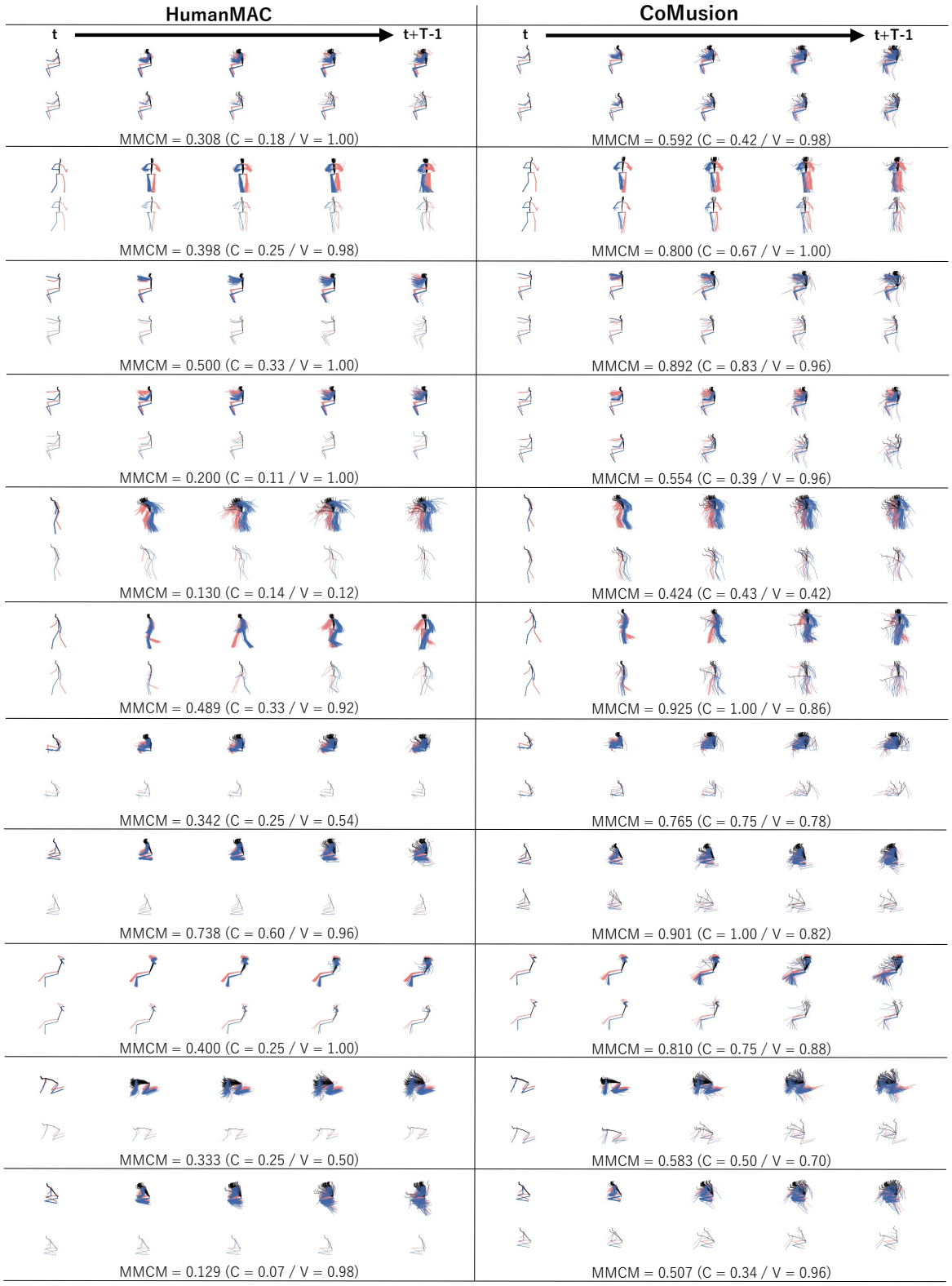


Figure 17. Multimodality comparison between HumanMAC and CoMusion, which are additional results of Fig. 4 in the main paper. The values of our metrics are displayed beneath each sequence.

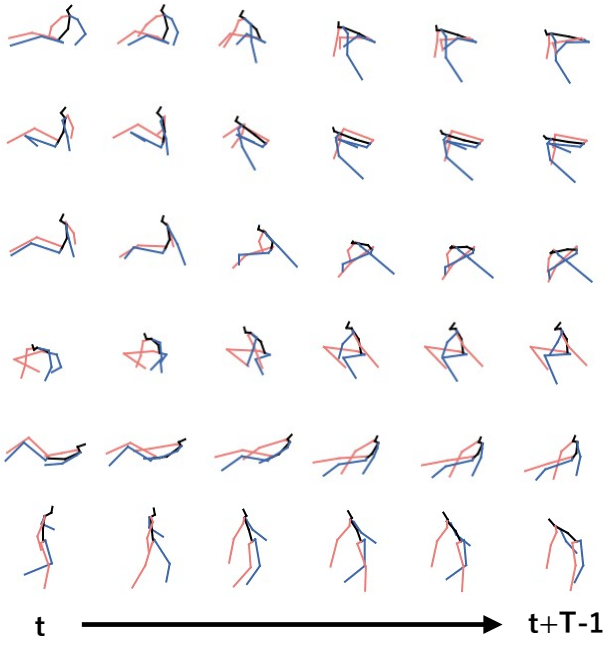


Figure 18. Predictions from DLow classified as abnormal, which are additional results of Fig. 5 in the main paper.

measures how closely the predicted distribution concentrates around a single GT, whereas MMCM evaluates the agreement between the predictions and the multiple GTs (MMGTs). Hence, different outcomes between MMCM and NLL are to be expected. As FID and CMD are realism-aware metrics, they serve a different purpose from MMCM's assessment of multimodality and therefore exhibit different trends from those of MMCM.

H. Explicit Algorithm

In this section, we show the explicit algorithm of MMCM to help implement it. Algorithm 1 summarizes the whole process mentioned in Sec. 4 in the main paper.

I. Detailed Description of Distance Threshold

To remove the outlier, we use a distance threshold τ for abnormality: if the distance to the nearest mode exceeds τ, the prediction is labeled to be abnormal. As explained in the main paper, τ is empirically set larger than the maximum distance observed when the same procedure is applied to only normal motions, thereby preventing false positives. The resulting values are $\tau = 1.024$ for H36M and $\tau = 3.140$ for AMASS.

A potential concern is that MMCM might be highly sensitive to the choice of τ: if so, its scores could vary widely across datasets or even reverse the ranking of methods, undermining MMCM's reliability. To verify robustness, we vary τ and plot MMCM for three representative methods (DLow [24], BeLFusion [4], and CoMusion [22]); the curves are shown in Fig. 19. For both datasets, MMCM converges once τ exceeds a certain value, and our selected thresholds lie in this plateau region. Thus, MMCM is not largely dependent on τ setting, and moderate changes to the threshold have little impact on the scores.

These results indicate that the chosen τ removes only highly abnormal motions. In other words, most abnormal predictions are not identified by the threshold τ, but rather

Algorithm 1: Computation of MMCM

Input: Last 3 frames of past $\mathbf{X}' = \{p_{t-3}, p_{t-2}, p_{t-1}\}$, predicted future motions $\{\hat{\mathbf{Y}}^i\}_{i=1}^I$

Data: Cluster centres $\{\mu_k\}$, Modes of MMGTs $\mathbf{M}$, Distance threshold for abnormality τ

1. Mode assignment

$\mathbf{d}, \mathbf{k} \leftarrow \text{NearestCentre}(\{\mathbf{X}' + \hat{\mathbf{Y}}^i\}_{i=1}^I, \{\mu_k\})$

for $i = 1, \ldots, I$ **do**

$$\hat{m}_i \leftarrow \begin{cases} k_i & \text{if } d_i \le \tau \\ -1 & \text{otherwise} \end{cases}$$

$\hat{\mathbf{M}} \leftarrow \{\hat{m}_1, \ldots, \hat{m}_I\}$

2. Mode Coverage Rate C**, Mode Validity Rate** V

$U, \widehat{U} \leftarrow \text{unique}(\mathbf{M}), \text{unique}(\hat{\mathbf{M}})$

$$C \leftarrow \frac{|U \cap \widehat{U}|}{|U|}$$

$$V \leftarrow \frac{|\{\hat{m}_i \in U\}|}{|\mathbf{M}|}$$

3. Score

$$\text{MMCM} \leftarrow \frac{2\,C\,V}{C + V}$$

return MMCM

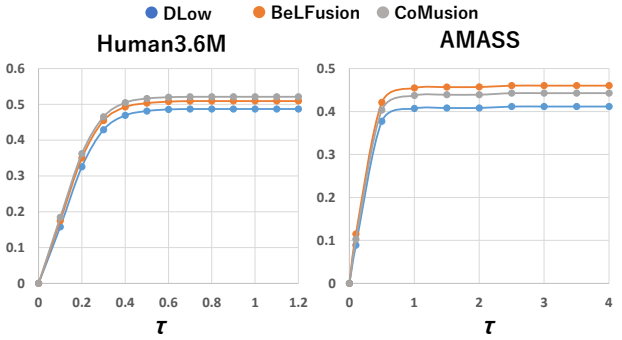


Figure 19. MMCM sensitivity to the threshold τ. For both H36M (left) and AMASS (right), the scores for DLow, BeLFusion, and CoMusion stabilise once τ exceeds a certain value. The chosen τ (1.024 in H36M and 3.140 in AMASS) is in this plateau region.

Table 9. Comparison with "MMCM without kinematic constraint (original)" and "MMCM with kinematic constraint", respectively described as "w/o Kin. Const." and "w/ Kin. Const.". Red and blue indicate the best and second-best results for each type.

Method	MMCM↑(C↑/V↑)	
	w/o Kin. Const.	w/ Kin. Const.
DLow	0.487 (0.569/0.539)	0.274 (0.412/0.282)
HumanMAC	0.492 (0.465/0.713)	0.412 (0.425/0.562)
CoMusion	0.521 (0.528/0.666)	0.423 (0.477/0.503)

by whether they fall outside the set of valid modes determined after mode assignment.

J. Kinematic Constraints

In the original MMCM, a prediction is considered valid if its mode matches one of the MMGT modes in a fully data-driven fashion. This MMCM can be extended with explicit kinematic constraints. As a simple yet reasonable constraint in our preliminary experiments, we compare limb lengths (arms or legs) between the last frame of the past motion and the predicted motion. If any limb length changes by 25% or more, the motion is classified as kinematically invalid.

As shown in Table 9, the relative ranking remains consistent for all methods, regardless of whether the kinematic constraint is used. However, we also observe that the validity score V in MMCM decreases when the constraint is introduced, suggesting that the kinematic constraint correctly penalizes invalidly-changed limb lengths. These findings indicate that kinematic constraints can facilitate a more robust validity assessment. As future work, more principled constraints, such as anatomical priors inspired by recent studies [11, 18, 23, 25], could be designed to better capture the validity.

K. MMCM for 2D Trajectory Prediction

In this section, while MMCM is originally proposed for HMP, we evaluate 2D trajectory prediction methods by MMCM to demonstrate its applicability to other tasks.

K.1. Method

The MMCM for 2D trajectory prediction is computed in essentially the same way as the MMCM for HMP described in the main paper. The only difference lies in dimensionality reduction. For HMP, which involves high-dimensional data (approximately 5,000 dimensions), MMCM employs a two-stage pipeline with an autoencoder followed by UMAP. In contrast, since 2D trajectory data is low-dimensional (approximately 20 dimensions), we apply only UMAP. Apart from this point, it is identical to the original MMCM.

K.2. Experimental Setup

Datasets. These experiments are conducted on ETH/UCY dataset [10, 19]. This dataset consists of five subsets: ETH, HOTEL, UNIV, ZARA1, and ZARA2. Following common practice [9, 13], we set the past window to $B = 8$ frames and the prediction window to $T = 12$ frames.

Hyper parameters. As with MMCM for HMP, we tune its hyperparameters for each subset.

Metrics. In addition to MMCM, we also report APD—although it is rarely used in 2D trajectory prediction—to enable comparison with MMCM. We further include ADE and FDE, the widely used accuracy metrics for 2D trajectory prediction.

2D trajectory prediction methods. Metrics are evaluated with the following existing 2D trajectory methods: SingularTrajectory [3], FlowChain [13], MID [8], and Social-LSTM [1].

K.3. Experimental Results

Quantitative results for each method are showed in Table 10. Consistent with the HMP results, in 2D trajectory prediction, the MMCM scores exhibit trends that differ from APD and the accuracy metrics. This indicates that MMCM captures multimodality that prior metrics could not measure.

MMCM thus appears to have the potential to extend beyond HMP to other prediction tasks.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: human trajectory prediction in crowded spaces. In *CVPR*, 2016. 9

[2] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *SODA*, 2007. 4

[3] Inhwan Bae, Young-Jae Park, and Hae-Gon Jeon. Singulartrajectory: Universal trajectory predictor using diffusion model. In *CVPR*, 2024. 9

[4] Germán Barquero, Sergio Escalera, and Cristina Palmero. Belfusion: Latent diffusion for behavior-driven human motion prediction. In *ICCV*, 2023. 4, 8

[5] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Commun. Stat., Theory Methods*, 3(1), 1974. 5

[6] Lingwei Dang, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. Diverse human motion prediction via gumbel-softmax sampling from an auxiliary space. In *ACM MM*, 2022. 4

[7] David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE T-PAMI*, 1(2), 1979. 5

[8] Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *CVPR*, 2022. 9

Table 10. Quantitative comparison. C and V denote the average mode coverage rate and the average mode validity rate, respectively. Red and blue indicate the best and second-best results for each metric.

Subset	Metrics	SingularTrajectory	FlowChain	MID	Social-LSTM
ETH	MMCM↑ (C↑/V↑)	0.221 (0.401/0.212)	0.193 (0.326/0.186)	0.242 (0.370/0.290)	0.078 (0.050/0.388)
	APD ↑	4.808	4.049	2.879	0.000
	ADE ↓	0.413	0.545	0.531	1.185
	FDE ↓	0.671	0.987	0.841	2.435
HOTEL	MMCM↑ (C↑/V↑)	0.568 (0.597/0.671)	0.608 (0.551/0.905)	0.606 (0.585/0.805)	0.395 (0.330/0.911)
	APD ↑	5.020	1.340	2.561	0.000
	ADE ↓	0.121	0.187	0.182	0.392
	FDE ↓	0.186	0.351	0.282	0.785
UNIV	MMCM↑ (C↑/V↑)	0.371 (0.248/0.864)	0.244 (0.146/0.922)	0.292 (0.183/0.899)	0.027 (0.014/0.898)
	APD ↑	5.542	1.749	2.347	0.000
	ADE ↓	0.252	0.350	0.309	0.728
	FDE ↓	0.443	0.720	0.599	1.509
ZARA1	MMCM↑ (C↑/V↑)	0.430 (0.398/0.524)	0.493 (0.395/0.766)	0.437 (0.359/0.661)	0.066 (0.035/0.831)
	APD ↑	5.438	2.116	3.367	0.000
	ADE ↓	0.209	0.230	0.241	0.470
	FDE ↓	0.370	0.438	0.434	1.024
ZARA2	MMCM↑ (C↑/V↑)	0.667 (0.577/0.976)	0.610 (0.494/0.993)	0.607 (0.500/0.987)	0.422 (0.360/0.993)
	APD ↑	4.700	3.612	2.447	0.000
	ADE ↓	0.165	0.197	0.183	0.400
	FDE ↓	0.282	0.418	0.340	0.881

[9] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: socially acceptable trajectories with generative adversarial networks. In *CVPR*, 2018. 9

[10] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. *Comput. Graph. Forum*, 26:655–664, 2007. 9

[11] Ruotong Li, Weixin Si, Michael Weinmann, and Reinhard Klein. Constraint-based optimized human skeleton extraction from single-depth camera. *Sensors*, 19(11):2604, 2019. 9

[12] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, 28(2), 1982. 4

[13] Takahiro Maeda and Norimichi Ukita. Fast inference and update of probabilistic density estimation on trajectory prediction. In *ICCV*, 2023. 9

[14] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. Generating smooth pose sequences for diverse human motion prediction. In *ICCV*, 2021. 4

[15] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *JOSS*, 2(11), 2017. 1

[16] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. UMAP: uniform manifold approximation and projection. *JOSS*, 3(29), 2018. 1

[17] Fionn Murtagh and Pierre Legendre. Ward's hierarchical clustering method: Clustering criterion and agglomerative algorithm. *arXiv preprint arXiv: 1111.6285*, 2011. 4

[18] 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. 9

[19] Stefano Pellegrini, Andreas Ess, and Luc Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *ECCV*, 2010. 9

[20] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20, 1987. 5

[21] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *ECCV*, 2020. 6

[22] Jiarui Sun and Girish Chowdhary. Comusion: Towards consistent stochastic human motion prediction via motion diffusion. In *ECCV*, 2024. 4, 8

[23] Shashank Tripathi, Omid Taheri, Christoph Lassner, Michael J. Black, Daniel Holden, and Carsten Stoll. HUMOS: human motion model conditioned on body shape. In *ECCV*, 2024. 9

[24] Ye Yuan and Kris Kitani. Dlow: Diversifying latent flows for diverse human motion prediction. In *ECCV*, 2020. 4, 8

[25] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided human motion diffusion model. In *ICCV*, 2023. 9

[26] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. BIRCH: an efficient data clustering method for very large databases. In *ACM SIGMOD*, 1996. 4