RESEARCH-ARTICLE

# Joint-Dataset Learning and Cross-Consistent Regularization for Text-to-Motion Retrieval

**NICOLA MESSINA**, Institute of Information Science and Technologies, Pisa, Pisa, PI, Italy

**JAN SEDMIDUBSKY**, Masaryk University, Brno, Czech Republic

**FABRIZIO FALCHI**, Institute of Information Science and Technologies, Pisa, Pisa, PI, Italy

**TOMÁŠ REBOK**, Masaryk University, Brno, Czech Republic

**Open Access Support** provided by:

**Institute of Information Science and Technologies, Pisa**

**Masaryk University**



**Published:** 15 October 2025
**Online AM:** 12 June 2025
**Accepted:** 05 June 2025
**Revised:** 03 June 2025
**Received:** 27 June 2024

# Joint-Dataset Learning and Cross-Consistent Regularization for Text-to-Motion Retrieval

NICOLA MESSINA, CNR ISTI, Pisa, Italy
JAN SEDMIDUBSKY, Masaryk University, Brno, Czech Republic
FABRIZIO FALCHI, CNR ISTI, Pisa, Italy
TOMÁŠ REBOK, Masaryk University, Brno, Czech Republic

Pose-estimation methods enable extracting human motion from common videos in the structured form of 3D skeleton sequences. Despite great application opportunities, effective content-based access to such spatio-temporal motion data is a challenging problem. In this article, we focus on the recently introduced text-motion retrieval tasks, which aim to search for database motions that are the most relevant to a specified natural language textual description (*text-to-motion*) and vice-versa (*motion-to-text*). Despite recent efforts to explore these promising avenues, a primary challenge remains the insufficient data available to train robust text-motion models effectively. To address this issue, we propose to investigate joint-dataset learning—where we train on multiple text-motion datasets simultaneously—together with the introduction of a Cross-Consistent Contrastive Loss (CCCL) function, which regularizes the learned text-motion common space by imposing uni-modal constraints that augment the representation ability of the trained network. To learn a proper motion representation, we also introduce a transformer-based motion encoder, called MoT++, which employs spatio-temporal attention to process skeleton data sequences. We demonstrate the benefits of the proposed approaches on the widely used KIT Motion Language and HumanML3D datasets, including also some results on the recent Motion-X dataset. We perform detailed experimentation on joint-dataset learning and cross-dataset scenarios, showing the effectiveness of each introduced module in a carefully conducted ablation study and, in turn, pointing out the limitations of state-of-the-art methods. The code for reproducing our results is available here: https://github.com/mesnico/MOTpp.

CCS Concepts: • **Information systems → Retrieval models and ranking**; • **Computing methodologies → Visual content-based indexing and retrieval**;

Additional Key Words and Phrases: 3D human motion, cross-modal retrieval, multi-modal understanding, text-motion retrieval

Nicola Messina and Jan Sedmidubsky contributed equally to this research.

This research was supported by the Ministry of the Interior of the CR project "Automated digital data forensics lab for complex crime detection" (No. VK01010147), FAIR—Future Artificial Intelligence Research—Spoke 1 (EU NextGenerationEU PNRR M4C2 PE00000013), AI4Media—A European Excellence Centre for Media, Society, and Democracy (EC, H2020 No. 951911), and SUN—Social and hUman ceNtered XR (EC, Horizon Europe No. 101092612).

Authors' Contact Information: Nicola Messina (corresponding author), CNR ISTI, Pisa, Italy; e-mail: nicola.messina@isti.cnr.it; Jan Sedmidubsky, Masaryk University, Brno, Czech Republic; e-mail: sedmidubsky@mail.muni.cz; Fabrizio Falchi, CNR ISTI, Pisa, Italy; e-mail: fabrizio.falchi@isti.cnr.it; Tomáš Rebok, Masaryk University, Brno, Czech Republic; e-mail: rebok@ics.muni.cz.



ACM 1551-6865/2025/10-ART289
https://doi.org/10.1145/3744565

**ACM Reference format:**
Nicola Messina, Jan Sedmidubsky, Fabrizio Falchi, and Tomáš Rebok. 2025. Joint-Dataset Learning and Cross-Consistent Regularization for Text-to-Motion Retrieval. *ACM Trans. Multimedia Comput. Commun. Appl.* 21, 10, Article 289 (October 2025), 24 pages.
https://doi.org/10.1145/3744565

## 1 Introduction

Pose-estimation methods [9] reconstruct the virtual 3D positions of human-body keypoints from a single-camera video stream. The positions of keypoints estimated in individual frames constitute a simplified spatio-temporal representation of human motion known as a *skeleton sequence*. Analyzing human motion through this skeleton modality offers several advantages over the video modality [60], including higher motion abstraction, much less sensitivity to background information, more consistent representation across different viewpoints, and much better computational efficiency [36]. As indicated in [46], the analysis of the skeleton representation unlocks unprecedented application potential in many domains. For example, the skeleton data could be used in telemedicine to remotely evaluate a patient's progress in rehabilitation, in sports to assess a figure-skating performance without the emotions of human referees, in smart cities to detect potential threats from surveillance cameras like a running group of people, in virtual reality to transpose real movements into virtual environments, or in robotics to study and develop humanoid robots and human-robot interfaces [46]. The ever-increasing popularity of skeleton data calls for technologies able to semantically analyze large volumes of such spatio-temporal data with respect to the focus of a target application.

Current research in skeleton data analysis primarily focuses on designing deep-learning architectures for classification of actions [59, 61, 76] or detection of such actions in continuous streams [12, 38, 57]. The trained network architectures can then serve as *motion encoders* that express the motion semantics by a high-dimensional *embedding* (i.e., *feature vector*) extracted from the last hidden network layer. To compute the similarity between a pair of motions, the distance between their embeddings is calculated, e.g., using the cosine or Euclidean distance functions. Traditional content-based motion retrieval methods [4, 21, 46] are based on the *query-by-example* paradigm, which aims at identifying the database motions that are the most similar to a user-defined query motion example. However, specifying a convenient query motion example may be problematic or even impossible since such an example might not ever exist.

These challenges inspire the development of smarter techniques for understanding and accessing 3D motion data. In this article, we concentrate on learning a latent interaction among motion and text modalities, focusing on the challenging *text-to-motion retrieval* task—which aims at searching a database of skeleton sequences and determining those that are the most relevant to a textual query formulated in natural language description—as well as its symmetric *motion-to-text retrieval* variant, which aims at finding the text sentences in a database mostly relevant to a given query motion, as illustrated in Figure 1. Specifically, text-to-motion retrieval has nice downstream applications for efficiently and effectively browsing large motion collections without relying on the query-by-example paradigm. This task was introduced in some recent works [36, 41] and is still relatively underexplored. The objective of the original idea [36] is to separately encode motion and text modalities and project the obtained motion and text latent representations into the same common space. We build upon this idea [36] and incorporate some common retrieval benchmarking protocols employed in [41]. We also introduce an improved transformer-based motion encoder called MoT++, as the extension of **Motion Transformer (MoT)** proposed in [36]. One of the key contributions of this article is to solve a principal problem arising in this domain, which is the lack of sufficient data

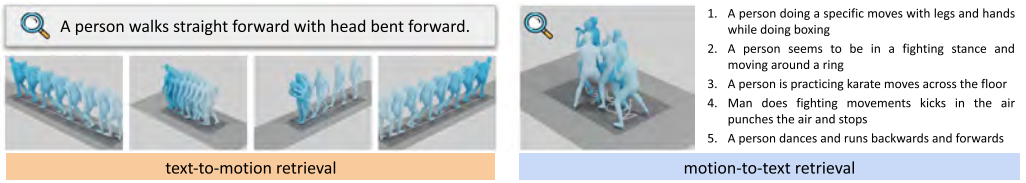



Fig. 1. Formulation of the tasks: text-to-motion retrieval (left) and motion-to-text retrieval (right).

to obtain good motion-to-text and text-to-motion retrieval generalization. To achieve this goal, we progress toward scenarios that employ the two key datasets in this domain as training data in a *joint-dataset learning (JDL)* setup. Furthermore, we enhance the training process with a new loss function, which we call **Cross-Consistent Contrastive Loss (CCCL)**, that regularizes the learned embedding space by imposing some cross-consistency constraints among the scores computed within and across modalities to improve generalization.

## 2 Related Work

For the adopted text-to-motion retrieval task, we mainly need to: (1) encode both text and skeleton data modalities into compact and content-preserving latent representations, (2) learn the common multi-modal space for both the modalities, and (3) manage the motion representations so that the most relevant ones are efficiently obtained for a given text representation. Therefore, we present a survey of existing methods related to: (i) text encoders, (ii) motion encoders, and (iii) cross-modal techniques processing both the motion and text modalities.

### 2.1 Text Encoders

Recent advancements in neural language models strongly help solve various tasks in natural language processing. The mainstream is to pre-train Transformer models over large-scale corpora. The resulting models differ in size, ranging from smaller (e.g., BERT [22] with 110 million parameters) to large language models (e.g., GPT-3 [3] with 175 billion parameters). Since the domain of human motion is particularly limited, it is sufficient to employ small-scale models. In [13], a BERT-based language model is employed within the motion synthesis task conditioned on a natural language prompt. This model stacks together a BERT pre-trained module and an LSTM model composed of two layers for aggregating the BERT output tokens, producing the final text embedding. In the text-to-motion retrieval task [36], the final hidden state of the LSTM model is considered as the final sentence representation. CLIP [43] is the vision-language model trained in a contrastive manner for projecting images and natural language descriptions in the same common space. The textual encoder of this model is composed of a transformer encoder [58] with modifications introduced in [44] and employs lower-cased byte pair encoding representation of the text. This encoder is also utilized in text-to-motion retrieval [36] by stacking an affine projection to the CLIP representation. In [41], the authors employ a simple transformer-based encoder inspired by the **Variational Autoencoder (VAE)** encoder in [39]. This encoder, called *ACTORStyleEncoder*, takes input text sequences of arbitrary length concatenated to two learnable `[class]` tokens. In output, it employs these two special tokens to estimate the mean and variance vectors of a multinomial Gaussian distribution, used by the decoder to reconstruct the corresponding motion. Despite not being pre-trained on large textual corpora, ACTORStyleEncoder seems better able to capture motion-related textual descriptions, outperforming CLIP in this domain. For this reason, we also employ such an encoder to obtain a suitable textual representation.

### 2.2 Motion Encoders

Motion encoders have traditionally learned latent motion representations in a supervised way for the classification task, mainly based on transformers [1, 6, 7, 36, 61], convolutional [32], recurrent [51], or graph-convolutional [8, 76] networks, or their combinations (e.g., transformer and 3D-CNN [49]). The current trend is to rely on self-supervised methods [17, 52, 53, 59], as they can learn motion semantics without knowledge of labels using reconstruction-based or contrastive-based learning. The reconstruction-based approach [45, 68] applies the encoder-decoder principle to reconstruct the original skeleton data of an input motion and uses the learned intermediate feature as the latent representation. The contrastive-learning approach [17, 28, 65] aims at learning a meaningful metric that sufficiently reflects semantic similarity to discriminate motions belonging to different classes in the validation step.

To increase the descriptive power of the learned latent representations, the learning process can integrate other skeleton modalities that are extracted from 3D coordinates and provide complementary information to the original joint modality, as recently surveyed in [47]. For example, early-fusion strategy in [53] is applied to jointly encode the joint, bone, and motion modalities in a single-stream manner. In [5], the late-fusion strategy is applied to skeleton modalities of joints and bones and also to the motion-map modality extracted from RGB frames cropped around the detected skeletons. In [26], RGB and depth streams are encoded using a recurrent neural network with specialized multi-modal contextualization units. In [12], a fusion module of skeleton and RGB modalities is proposed to enable the two features to guide each other.

We primarily focus on two recent motion encoders proposed in [36, 41]. In [41], the authors employ the same ACTORStyleEncoder not only for text encoding but also for motion encoding, as originally proposed in [39]. In the case of the motion modality, input tokens representing words in a phrase are replaced by skeleton poses across different timesteps. The motion encoder in [36] introduces a spatio-temporal transformer, called MoT, which is principally similar to ViViT for video encoding [2]. MoT processes both spatial and temporal features using the attention mechanism. In this work, we improve MoT by employing a different kind of spatio-temporal attention and a methodology to aggregate skeleton joints without losing feet and root information.

### 2.3 Mutual Processing of Motion and Text Modalities

The current trend in multimedia processing is to learn a common multi-modal space for the visual and textual modalities [10, 29, 35, 43, 48] so that similar images or videos can be described and searched with textual descriptions. This enables the use of open vocabularies or complex textual queries to search for relevant images/videos. Inspired by such powerful and versatile text-vision models, new works [14, 24, 39, 40, 56, 68, 69, 71] have started to emerge also in the *text-motion* domain. These works focus on *motion generation*, i.e., generating skeleton avatars from a textual description. In contrast to video data, the skeleton modality is anonymized and avoids learning many common biases present in video datasets. The principal idea of these text-driven motion generation methods is the same as the text-vision methods—to align text and motion embeddings into the common space, as successfully implemented by these seminal approaches: MDM [56], MotionDiffuse [69], and T2M-GPT [68]. Both the MDM [56] and MotionDiffuse [69] methods employ the diffusion principle to gradually add noise to a sample from the data distribution and learn the reverse process of denoising the sample by a backbone generative neural network. This can be further enhanced by predicting a proper motion length from text [14] or by integrating multi-modal inputs (e.g., text and single-frame poses) [71]. The text-to-motion task can also be trained jointly with the reverse motion-to-text task [15], which enables *motion captioning*, i.e., generating a text

description for an input motion, as successfully introduced by TM2T [15]. This captioning approach is enhanced by MotionGPT [71] that transforms the motion modality into discrete tokens—similarly as text into text tokens—and performs language modeling on both motion and text in a unified manner.

Besides motion generation, text-motion processing has been used for improving skeleton-based *classification* by text-to-motion matching [23] or by additionally generated textual descriptions from training actions [61]. Employing the text-motion modalities for retrieval purposes has not yet been studied much. There are two fundamental papers [36] (SIGIR 2023) and [41] (ICCV 2023) that independently tackle the text-motion retrieval tasks. Both approaches are essentially very similar as they learn a common cross-modal embedding space for both the text-motion training pairs with the InfoNCE loss function. While the former approach [36] proposes an MoT to encode motions, the latter approach [41] adopts ACTORStyleEncoder trained for motion synthesis using contrastive learning with careful selection of negatives by filtering out the *wrong* negatives—which are the negative samples too similar to the positive one. In the retrieval phase, the embedding of a user text query is extracted and compared to the embeddings of database motions to determine the $K$-nearest-neighbor motions. Both the approaches achieve competitive retrieval results on the **KIT Motion Language (KITML)** [42] and HumanML3D [14] benchmark datasets.

Other subsequent text-motion retrieval works [11, 62–64, 66, 67] have recently emerged. In [67], an image-patch-based motion representation is proposed to adopt varying skeleton structures. In [63], the common space properties are learned more effectively by carefully selecting hard negative samples during triplet-loss training. In [64], the hardest negative samples are mined in combination with Max of Hinges Loss, since standard triplet loss can lead to local minima when many negative samples are close to the anchor but violate hard constraints. In [11], negative samples include texts with a shuffled sequence of events to better align text and motion modalities. To capture local and subtle motion variations, local text and motion representations are assigned into a set of cross-modal local aggregated descriptors [66]. In [62], the retrieval concept is enriched by the ability of localization of start and end timestamps of query-relevant subsequences within an untrimmed skeleton sequence.

*Contributions of This Article.* The two mainstream text-to-motion retrieval approaches [36, 41] learn the text-motion representations independently for each dataset, either KITML or HumanML3D. Although *cross-dataset* (i.e., training dataset differs from validation one) or *JDL* (i.e., training and validation datasets come from the fusion of multiple diverse datasets) learning approaches are principally known [54], they have not been tackled in the text-motion domain. Studying the generalization abilities of retrieval methods in low-data regimes is of critical importance for achieving good and reliable models.

In light of the above, we contribute to the field in two principal ways: (1) by proposing MoT++, an improvement of MoT [36], which includes a more effective spatio-temporal attention mechanism and a more nuanced joint aggregation schema, and (2) by tackling the lack of motion-text data by studying *joint-dataset* and *cross-dataset* generalization, together with a new loss function that regularizes the learned common space by also enforcing uni-modal score constraints. Our contributions can be summarized as follows:

—We introduce motion encoder MoT++ that improves on MoT [36] by integrating effective spatio-temporal attention schemas and preserving information about feet and root joints better.

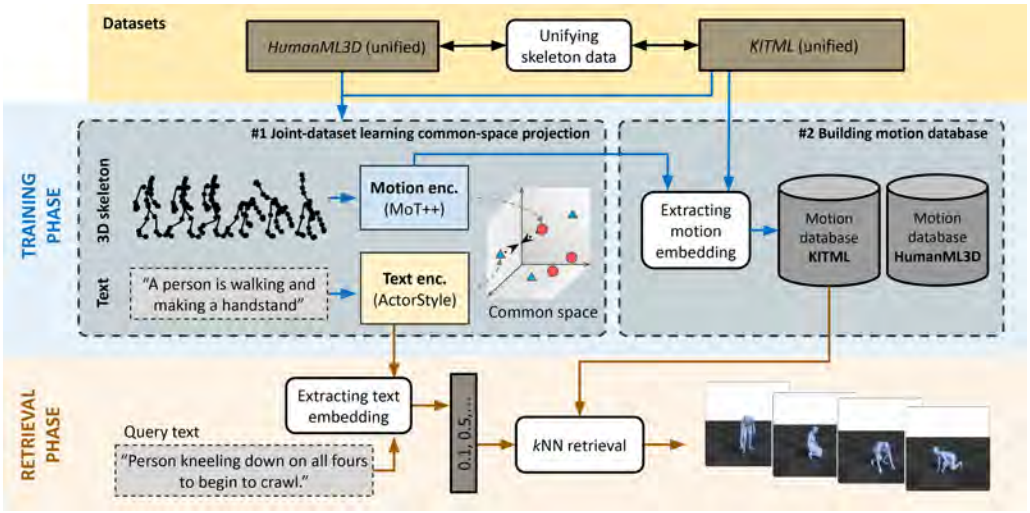



Fig. 2. Schematic illustration of the whole architecture. In the training phase, (1) JDL is applied to unified HumanML3D and KITML datasets to learn the common space of both the text and motion modalities and (2) the trained motion encoder is then used to extract motions' embeddings that are stored in a database (exemplified on the KITML dataset). In the retrieval phase, the embedding of a given text query is extracted and compared against the embeddings of motions in the KITML database to retrieve the $K$ most relevant motions.

- We explore *cross-dataset* evaluation and JDL to understand and mitigate generalization issues due to the lack of sufficiently diverse data.
- We propose a new loss function, called CCCL, that better constrains the common space by augmenting the InfoNCE objective with uni-modal loss terms to mitigate the data scarcity issue more effectively.
- We perform an extensive experimental evaluation on single-dataset, cross-dataset, and JDL scenarios. We establish new baselines, especially for cross-dataset and JDL.
- We experimentally compare our approach with state-of-the-art methods using benchmarks defined for the KITML and HumanML3D datasets and present initial baseline results on the recent Motion-X dataset [27].

## 3 Text-to-Motion and Motion-to-Text Learning Pipeline

Our text-to-motion retrieval approach is principally inspired by state-of-the-art works in this field [36, 41]. In particular, it is a two-stream learning pipeline, where text and motion features are first extracted through *ad hoc* encoders and then projected into the same common space. We especially focus on JDL to achieve better generalizability of the learned features, as schematically illustrated in Figure 2. In this section, we sketch the components of the whole architecture and primarily focus on our contributions with respect to both state-of-the-art works, especially the new motion encoder, loss function, and JDL.

### 3.1 Problem Definition

We are given a database of $N$ text-motion pairs, defined as $\mathcal{S} = \{T_i, M_i\}_{i=1}^{N}$, where $M_i$ is the $i$th motion and $T_i$ its corresponding textual description specified in natural language, like "*A person kneeling down on all fours to begin to crawl.*" With such data organization, two symmetric tasks

are defined. In *text-to-motion* retrieval, a $T_i \in \mathcal{S}$ is used as a query to retrieve the $K$ most relevant motions $\{M_j\}_{j=1}^{K}$ from $\mathcal{S}$ based on how closely they align with the semantics of text. Conversely, in the symmetric *motion-to-text* retrieval task, a query motion $M_i \in \mathcal{S}$ is used to search for the most relevant textual descriptions $\{T_j\}_{j=1}^{K}$ from $\mathcal{S}$.

These retrieval tasks involve extracting the query feature using a proper motion or text encoder and then comparing it against the pre-extracted features of all database motions based on a predefined similarity function, such as the cosine similarity. This formulation, employed in many cross-modal retrieval scenarios [10, 20, 31, 36, 37, 41, 43], is very effective and efficient. In fact, if the motion database is very large, any vector-based indexing technique, such as FAISS,[1] can be principally adopted to speed up the motion retrieval process.

In this setup, we construct a probabilistic common embedding space as introduced in [41]. Specifically, we extract the following quantities from each $T_i$ and $M_i$ in $\mathcal{S}$: $\mathbf{m}_i^{\mu}, \mathbf{m}_i^{\sigma^2} = \mathcal{E}_m(M_i)$ and $\mathbf{t}_i^{\mu}, \mathbf{t}_i^{\sigma^2} = \mathcal{E}_t(T_i)$, where $\mathcal{E}_m$ and $\mathcal{E}_t$ are two deep neural networks having their own trainable parameters that digest motions $M_i$ and texts $T_i$, respectively. These networks output two vectors each. Specifically, $\mathbf{m}_i^{\mu}, \mathbf{m}_i^{\sigma^2}$ represent the mean and variance of a Gaussian distribution living in the common space and representing motion $M_i$, while $\mathbf{t}_i^{\mu}, \mathbf{t}_i^{\sigma^2}$ plays the same role for text $T_i$. This formulation is driven by the underlying VAE [25] core structure of the motion pipeline, which tries to reconstruct the original motion $\tilde{M}_i$ through a decoder $\mathcal{D}_m$, computing $\tilde{M}_i = \mathcal{D}_m(\mathbf{z})$ where $\mathbf{z} \sim \mathcal{N}(\mathbf{m}_i^{\mu}, \mathbf{m}_i^{\sigma^2})$. The losses employed to constrain the latent space in the VAE include the standard **Kullback-Leibler (KL)** divergence loss—that forces both text and motion features to share similar distributions and force them to the unit normal distribution—and a motion reconstruction loss employed on the decoder's head—formalized by a per-joint regression objective $\mathcal{L}_{\text{rec}} = \sum_i ||M_i - \tilde{M}_i||_1$—which in this setup plays only a regularization role by stabilizing the whole training process. In fact, as our downstream task is *retrieval* and not *generation*, we are not interested in the outcome of the motion generation part of the network, although it has been shown to produce nice regularization effects. Instead, the core objective for text-to-motion retrieval and motion-to-text retrieval is the creation of a common space where motions and corresponding descriptions have high similarity, while motions and uncorrelated descriptions have low similarity. This contrastive objective is enforced by a loss $\mathcal{L}_{\text{retrieval}} = \text{contrastive}(\{\mathbf{m}_i^{\mu}, \mathbf{t}_j^{\mu}\}_{i,j=1}^{N})$ which, in our proposal, will be detailed in Section 3.3. Notice that, as in [41], only mean vectors are used as features for the motion and text in the common space instead of their Gaussian distributions used to reconstruct motions. For this reason, from this point onward, we simply refer to the $i$th text and motion features without the $\mu$ notation, as $\mathbf{m}_i$ and $\mathbf{t}_i$.

### 3.2 MoT++ Motion Encoder

One of the key contributions of this article is the proposed transformer-based motion encoder MoT++, which extends its predecessor MoT presented in [36]. The architecture of MoT++ is built on top of the successful transformer-based video processing network ViViT [2], which implements efficient spatio-temporal attention mechanisms, namely *factorized encoder* and *factorized self-attention* to factorize attention computation in either the spatial and temporal dimensions, drastically reducing the needs of memory and computational resources. In the following two subsections, we present the architecture behind MoT++ and some details behind the processing of the skeleton features used to feed it.

[1] https://ai.meta.com/tools/faiss/.

*3.2.1 Architecture.* MoT++ takes as input a raw motion sequence $\bar{\mathbf{x}} \in \mathbb{R}^{T \times J \times D}$, where $T$ is the number of frames of the motion, $J$ is the number of skeleton joints, and $D$ is the dimensionality of each joint.

First, the input is pre-processed by a function $\mathcal{J}(\bar{\mathbf{x}})$ which possibly aggregates joints together, reducing $J$ to $J'$ and increasing their dimensionality from $D$ to $D'$ to better fit the transformer pipeline. We can consider each of the $J'$ new elements as a group of skeleton joints, each represented by a new $D'$-dimensional feature. A new sequence of $J' < J$ joints serves two purposes: (i) it is beneficial from a computational point of view, as the spatial sequence in input to the transformer is shorter, and (ii) it helps in reducing overfitting and increasing generalization of the whole pipeline. As the $\mathcal{J}$ function, we employ a set of independent MLPs, which separately aggregate joints from seven different parts of the skeleton, following an idea similar to [13] and further developed in the previous MoT framework [36]. Specifically, the first five groups are obtained by aggregating joints from the five different parts of the human body. We then include two separate MLPs for independently processing the root bone and feet floor contact state. At the end of this step, we obtain the pre-processed motion $\mathbf{x} = \mathcal{J}(\bar{\mathbf{x}}) \in \mathbb{R}^{T \times J' \times D'}$.

The resulting sequence $\mathbf{x}$ is flattened into $\mathbb{R}^{TJ' \times D'}$ and appended to two CLS tokens $\text{CLS}_\mu$ and $\text{CLS}_{\sigma^2}$ used to follow the same encoder output interface employed by the TMR framework [41], which estimates mean and variance of the latent space. We therefore obtain a pre-processed sequence $\mathbf{x}_s = [\text{CLS}_\mu, \text{CLS}_{\sigma^2}, \mathbf{x}] \in \mathbb{R}^{(2+TJ') \times D'}$ ready to be fed into the transformer layers. This input is processed through $N$ transformer layers, denoted by $\{\mathcal{T}_i\}_{i=1}^{N}$. Similarly to ViViT, each transformer layer processes either spatial information or temporal information, avoiding jointly processing space and time dimensions, which would be computationally unfeasible. For this reason, $N$ is always even so that we always have the same number of temporal and spatial layers.

In the so-called *factorized encoder* case, the first $N/2$ layers compute attention over the joint dimension to share information across skeleton joints at each timestep independently. From a practical perspective, this is obtained by simply reshaping $\mathbf{x}_s \in \mathbb{R}^{(TJ'+2) \times D'}$ into $\mathbf{x}_s^r \in \mathbb{R}^{T \times (2+J') \times D'}$, and then computing $\mathbf{x}_t = \{\mathcal{T}_i\}_{i=1}^{N/2}(\mathbf{x}_s^r)$, with the $T$ dimension playing the role of the batch size. The second $N/2$ stack of layers performs the same operation but on the time dimension for each joint. This is obtained by first permuting and reshaping $\mathbf{x}_t$ into $\mathbf{x}_t^r \in \mathbb{R}^{J' \times (2+T) \times D'}$ before seeding it into these layers, to obtain $\mathbf{x}_o = \{\mathcal{T}_i\}_{i=N/2+1}^{N}(\mathbf{x}_t^r)$, where this time $J'$ plays the role of the batch size.

The so-called *factorized self-attention*, employed in MoT [36], is very similar to the factorized encoder case, except that spatial and temporal transformer layers are interleaved: $\{\mathcal{T}_{2i}\}_{i=1}^{N/2}$ layers compute attention over time, while $\{\mathcal{T}_{2i-1}\}_{i=1}^{N/2}$ layers work out attention over the dimension of joints.

From early experimentation, we found that *factorized encoder* works better than *factorized self-attention* on the motion domain. For this reason, we instantiate MoT++ by employing the *factorized encoder*. The output of MoT++ is composed of the first two tokens of the output sequence $\mathbf{x}_o$, i.e., the content of the two prepended CLS tokens that are devoted to producing the mean and variance over the latent space. Figure 3 shows the overall architecture of the proposed MoT++ motion encoder.

*Motion Features.* One of the key advantages of MoT++ over other proposed encoders is that it employs a well-structured spatial sequence of joint tokens instead of a single flattened feature vector representing the whole skeleton. In fact, by design, MoT++ requires a 2D sequence of tokens (a spatial and a temporal one), which does not allow for a single flattened feature vector encoding the full skeleton. In order to achieve this, the most intuitive solution is to break down the flattened representation employed in previous works [40, 41, 56] to reconstruct a sequence of features, one for

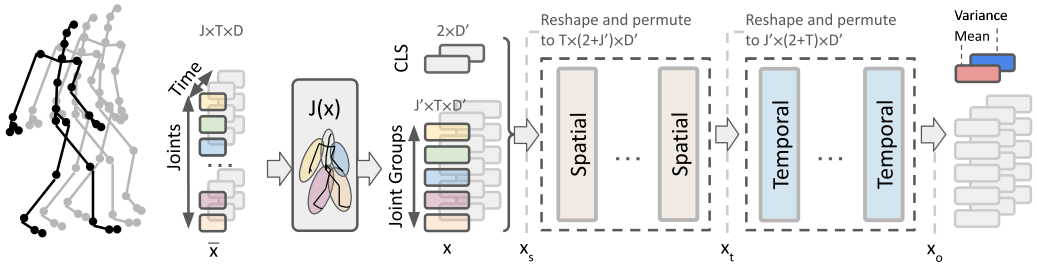



Fig. 3. MoT++ architecture. The input spatio-temporal skeleton sequence $\bar{\mathbf{x}}$ is processed by the $\mathcal{J}$ function to spatially group the skeleton joints and therefore reduce the spatial sequence length while increasing the dimensionality $D$ to $D'$. The resulting sequence $\mathbf{x}$, concatenated to two special tokens, is then processed by a spatio-temporal transformer—in this case configured in a *factorized self-attention* setup. The two CLS tokens in output are employed as $\mathbf{m}^{\mu}$ and $\mathbf{m}^{\sigma^2}$.

each skeleton joint. Specifically, we employ the motion vector as pre-processed by [41] and extract from it the $D$-dimensional feature vector for each of the $J$ different skeleton joints. Specifically, we represent each joint with $D = 12$ different features, consisting of (i) a 3D sub-vector for encoding the rotation-invariant forward kinematic (*rifke*) motion information—the spatial position of each joint without root rotation applied; (ii) a 6D sub-vector encoding the rotation information of each joint in the 6D-continuous representation format [73]; and (iii) a 3D sub-vector encoding the spatial velocities of each joint. This is performed for all the non-root joints of the SMPL skeleton [30], which corresponds to 21 joints. In addition, we also collect a root joint token (composed of rotation velocity along the $y$-axis, the linear velocity on the $xy$ plane, and the root height) and a feet token (a virtual token carrying foot contact information for a total number of $J = 23$ skeleton joints).

### 3.3 CCCL

Although the motion encoder is a core ingredient for achieving a clever comprehension of the underlying motion semantics, the role of data and of the optimization process are also of key importance. In particular, we noticed that, due to the scarcity of data, the networks are prone to overfitting if clever constraints on the learned common space are not imposed. To mitigate these issues, we propose to employ more than one dataset for training to increase the overall number of text-motion samples, which is anyway very small compared to data amount used in [34, 37, 43, 70], and introduce a novel loss that highly regularizes the produced common space to deal with limited data availability.

In this section, we introduce this novel loss, called CCCL. It is designed to impose additional constraints with respect to the standard multi-modal contrastive learning used in previous works to better constrain the training process and, in turn, achieve higher generalization. Specifically, given a text feature $\mathbf{t}_i$ and the corresponding motion feature $\mathbf{m}_i$, the proposed CCCL loss enforces the following objectives.

- —A cross-modal contrastive objective, which enforces $\mathbf{t}_i$ to have a higher cosine similarity to $\mathbf{m}_i$ with respect to $\mathbf{m}_j$ $(i \neq j)$ for text-to-motion retrieval and vice-versa for motion-to-text retrieval.
- —A uni-modal similarity objective for textual descriptions, which enforces semantically similar/dissimilar texts $\mathbf{t}_i$ and $\mathbf{t}_j$ $(i \neq j)$ to have high/low cosine similarity.
- —A uni-modal similarity objective for motion features, which enforces similar/dissimilar motions $\mathbf{m}_i$ and $\mathbf{m}_j$ $(i \neq j)$ to have high/low cosine similarity.

The idea behind CCCL is that the constraints imposed by the two uni-modal objectives help the standard cross-modal contrastive objective, by applying semantic constraints to the common space and, in turn, helping the generalization to different scenarios also in cases of data scarcity. In the following paragraphs, we provide better details on the cross- and uni-modal components of the proposed CCCL loss function.

*3.3.1 Cross-Modal Contrastive Objective.* The main optimization objective is the one that forces $\mathbf{t}_i$ and $\mathbf{m}_j$ to have maximum cosine similarity when $i = j$. Specifically, we employ InfoNCE loss, well-known from cross-modal matching in [72] and used, e.g., in text-image matching [43] or text-motion retrieval [36, 41]. InfoNCE is defined as:

$$\mathcal{L}_{\text{nce}} = -\frac{1}{B}\sum_{i}^{B} \log \frac{\exp(s(\mathbf{m}_i, \mathbf{t}_i)/\tau)}{\sum_j^B \exp(s(\mathbf{m}_i, \mathbf{t}_j)/\tau)} + \log \frac{\exp(s(\mathbf{m}_i, \mathbf{t}_i)/\tau)}{\sum_j^B \exp(s(\mathbf{m}_j, \mathbf{t}_i)/\tau)}, \tag{1}$$

where $\tau$ is a temperature parameter learned during training, $s(\cdot, \cdot)$ is the cosine similarity between two feature vectors, and $B$ is the batch size.

*3.3.2 Uni-Modal Similarity Objective.* Enforcing the cross-modal contrastive objective alone does not directly imply any strong semantic constraints between textual descriptions or motions. This means that if $\mathbf{t}_i$ is close to $\mathbf{m}_j$ in the common space, this does not directly imply that $\mathbf{t}_i$ is close to $\mathbf{t}_j$, and in the same way, we cannot assume that $\mathbf{m}_i$ is close to $\mathbf{m}_j$. This objective serves exactly to bridge this semantic gap in the two uni-modal domains.

We propose the constraint to align the score distribution of cross-modal matching to the two uni-modal domains. We employ the KL divergence objective to estimate the distance between these score distributions. Specifically, given the distributions of the scores $\mathcal{S}$ directly derived from the cross-modal and uni-modal cosine similarities:

$$\begin{aligned} \mathcal{S}_j^{\text{t2m}} &= \text{softmax}_i\ s(\mathbf{t_j}, \mathbf{m_i}); \quad \mathcal{S}_j^{\text{m2t}} = \text{softmax}_i\ s(\mathbf{t_i}, \mathbf{m_j}); \\ \mathcal{S}_j^{\text{m2m}} &= \text{softmax}_i\ s(\mathbf{m_i}, \mathbf{m_j}); \quad \mathcal{S}_j^{\text{t2t}} = \text{softmax}_i\ s(\mathbf{t_i}, \mathbf{t_j}), \end{aligned} \tag{2}$$

we define the following objectives:

$$\begin{aligned} \mathcal{L}_{\text{cross-to-m2m}} &= \frac{1}{B}\sum_{j}^{B} \frac{\text{SymmKL}(\mathcal{S}_j^{\text{t2m}}, \mathcal{S}_j^{\text{m2m}}) + \text{SymmKL}(\mathcal{S}_j^{\text{m2t}}, \mathcal{S}_j^{\text{m2m}})}{2} \\ \mathcal{L}_{\text{cross-to-t2t}} &= \frac{1}{B}\sum_{j}^{B} \frac{\text{SymmKL}(\mathcal{S}_j^{\text{t2m}}, \mathcal{S}_j^{\text{t2t}}) + \text{SymmKL}(\mathcal{S}_j^{\text{m2t}}, \mathcal{S}_j^{\text{t2t}})}{2}, \end{aligned} \tag{3}$$

where $\text{SymmKL}(\mathcal{X}, \mathcal{Y}) = \frac{\text{KL}(\mathcal{X}, \mathcal{Y}) + \text{KL}(\mathcal{Y}, \mathcal{X})}{2}$ is the symmetric version of the KL divergence. We use the symmetric version since there is no distribution among the involved ones that can be elected as the *reference* distribution, given that all these distributions can be mutually adjusted during the training phase.

Therefore, we can derive the first objective, which takes into account the score distribution similarity between cross- and uni-modal features: $\mathcal{L}_{\text{cross-to-uni}} = \mathcal{L}_{\text{cross-to-t2t}} + \mathcal{L}_{\text{cross-to-m2m}}$. However, this objective alone may be insufficient. In fact, there is an important training signal that is required to avoid degenerating into potentially trivial or semantically incorrect solutions. In particular, we need, at least in the warmup training phase, some external supervision providing us with meaningful uni-modal score distributions $\mathcal{S}^{\text{t2t}}$ and $\mathcal{S}^{\text{m2m}}$ to better guide the organization of motion or text features within the respective uni-modal manifolds in the common space.

To this aim, we employ teacher models able to provide guidance for $\mathcal{S}^{\text{t2t}}$ and $\mathcal{S}^{\text{m2m}}$. For the text, we can easily employ a textual model $\mathcal{E}_t^{\text{ref}}$ trained to estimate the similarity between two

sentences. Consequently, the text model works as a *teacher* model that guides the uni-modal scores distribution. Concerning motions, the situation is more challenging, given that motion classifiers [5] or motion autoencoder networks [45] able to derive meaningful comparable motion representations are usually trained on different skeleton formats and with diverse label distributions, which makes them not directly transferable to the datasets employed in this work. A naive yet reasonable solution to this problem is to assume that two motions are similar if their text descriptions are similar. In this way, if we assume $\mathcal{S}^{\text{textGT}} = \text{softmax}_i\ \mathcal{E}_t^{\text{ref}}(T_i, T_j)$ to be the scores distribution from the teacher text model, we can enforce the following:

$$\mathcal{L}^{\text{teacher-to-t2t}} = \text{KL}(\mathcal{S}^{\text{textGT}}, \mathcal{S}^{\text{t2t}}); \quad \mathcal{L}^{\text{teacher-to-m2m}} = \text{KL}(\mathcal{S}^{\text{textGT}}, \mathcal{S}^{\text{m2m}}), \tag{4}$$

where, at this time, we employ the standard non-symmetric KL loss since $\mathcal{S}^{\text{textGT}}$ is considered as the true reference distribution. Therefore, we can derive our second objective, which takes into consideration the distribution gap between an optimal teacher and the uni-modal distributions: $\mathcal{L}^{\text{teacher-to-uni}} = \mathcal{L}^{\text{teacher-vs-t2t}} + \mathcal{L}^{\text{teacher-to-m2m}}$.

*3.3.3 Final Objective.* We finally combine the previously introduced loss functions:

$$\mathcal{L} = \mathcal{L}_{\text{nce}} + \lambda \mathcal{L}_{\text{cross-to-uni}} + (1 - \lambda)\mathcal{L}^{\text{teacher-to-uni}}, \tag{5}$$

where $\lambda$ is a hyper-parameter that balances the contribution of the teacher scores (supervised scores distillation) with respect to the self-alignment between cross- and uni-modal scores (self-sustained scores distillation). The overall optimization methodology is rooted in the idea that in the beginning, the text teacher's supervision signal helps guide the uni-modal manifolds toward an appropriate configuration, while it becomes unnecessary (or even counterproductive) if it is kept active for the whole training duration. For this reason, we introduce a simple linear scheduling policy for $\lambda$ that swipes this parameter from 0 (full teacher supervision) to 1 (full self-sustained learning regime) across various training epochs, following the swipe function:

$$\lambda(t) = \text{clamp}_{[0,1]}\left(\frac{t - t_{\text{start}}}{t_{\text{end}} - t_{\text{start}}}\right), \tag{6}$$

where $\text{clamp}_{[0,1]}(\cdot)$ ensures that the output stays bounded between 0 and 1, $t$ is the current epoch, and $t_{\text{start}}$ and $t_{\text{end}}$ are the epochs at which the swipe starts and ends, respectively. We experimentally found that $t_{\text{start}} = 40$ and $t_{\text{end}} = 100$ obtain an optimal performance. Therefore, we perform our main experiments employing this configuration. We dedicate a paragraph in the ablation study (Section 4.3) to better explore the role of these hyper-parameters.

## 4 Experimental Evaluation

In this section, we briefly describe benchmark datasets and a methodology for the evaluation of text-to-motion retrieval, as used in state-of-the-art approaches. We then report the performance of the proposed JDL and cross-dataset learning and compare the results to prior work. We also present the ablation study of our approach to measure the effects of JDL, the usage of different loss functions, the selection of particular hyper-parameters, and some experiments on the recently introduced Motion-X [27] dataset. In addition, we provide visualization of selected results of text-to-motion retrieval to highlight the strengths/weaknesses of the proposed approach.

### 4.1 Datasets and Evaluation Methodology

We employ HumanML3D [14] and KITML [42] datasets that are widely used for 3D text-to-motion generation/retrieval. We additionally adopt the recent Motion-X dataset [27] for ablation studies. These datasets provide captured 3D skeleton data and several human-written textual descriptions for each motion. Since the dataset body models are slightly different, we employ the pre-processing

pipeline[2] which unifies skeleton data into a common motion-text representation. This pipeline also includes extracting rotation-invariant forward kinematics motion features, such as 3D rotations, velocities, or foot contacts, originally proposed in [14]. This unification allows us to evaluate JDL and cross-dataset learning scenarios.

*KITML Dataset.* It contains 3,911 recordings of full-body motion in the Master Motor Map form [55], along with textual descriptions for each motion. It has a total of 6,278 annotations in English, where each motion recording has one or more annotations that explain the action, like "*A human walks two steps forwards, pivots 180 degrees, and walks two steps back.*"

*HumanML3D.* It is essentially very similar to KITML. However, it is a more recent dataset developed by adding textual annotations to already-existing and widely used motion-capture datasets—AMASS [33] and HumanAct12 [16]. It contains 14,616 motions annotated by 44,970 textual descriptions. It also contains KITML, although the original text descriptions have been completely rewritten. For this reason, we consider it safe to merge KITML and HumanML3D without incurring in strong training-testing interferences.

*Evaluation Methodology.* We employ the same evaluation methodology introduced in [41]. In particular, *recall* at rank $k$ ($R@k \uparrow$) measures the percentage of times the correct label is among the top $k$ results, i.e., the higher, the better. Since the recall is evaluated for $k \in \{1, 2, 3, 5, 10\}$, the *Rsum* metric is additionally presented as the sum of recall values over individual settings of $k$. We also report *median rank* ($MedR \downarrow$), i.e., the lower, the better. The results are reported on the test set of the respective datasets, i.e., on unseen motions. As in [41], we evaluate not only *text-to-motion* retrieval but also the orthogonal task of *motion-to-text* retrieval. We report the results using the four evaluation protocols also employed in [41]: (i) "All," most similar to the protocol employed in [36], where all the text and motions are used as query and retrieval set; (ii) "All with threshold," similar to "All" but a motion is considered correct if its description matches the query text above a threshold (set to 0.95 as in previous works); (iii) "Dissimilar subset," where a subset of 100 text-motion pairs are chosen so that the distances among the sampled texts are maximized; (iv) "Small batches," which randomly selects batches of 32 text-motion pairs and reports their average performance. To give an overall assessment of the probed methods, we also report the average of all the metrics over different protocols in the last group of rows in the tables. Notice that, unless otherwise stated, the line "MoT++" in the result tables also integrates the proposed CCCL loss function, with $t_{\text{start}} = 40$ and $t_{\text{end}} = 100$ (see Section 4.3 for detailed ablations on these hyper-parameters).

*Implementation Details.* All the methods were evaluated on three independent runs, and their average values were reported. Concerning TMR, the original code was used to generate the results on the cross-dataset evaluation and on the JDL setups. All the methods have been trained for 250 epochs, with a learning rate of 5e-5, employing an RTX 2080Ti GPU. The text encoder, ACTORStyleEncoder, is the same as in TMR and was re-trained from scratch at every run. As the teacher text model $\mathcal{E}_t^{\text{ref}}$, we employed a pre-trained MPNet [50]. The probed Rehamot model was run both using the original **Cross-perceptual Salience Mapping (CPS)** matching function (Rmt in short), and also the standard cosine similarity (indicated as Rmt*), to better compare the method with TMR and MoT++. Concerning MoT++, we downsampled motions to 200 frames if longer than this amount. We employed 1024-D fully connected connections within the attention blocks with 4 heads. We used a factorized encoder configuration, with two spatial layers followed by two temporal ones. As in previous works [36, 41], we set the size of the common embedding space to 256.

[2] https://github.com/Mathux/AMASS-Annotation-Unifier.

Table 1. JDL: Training on KITML+HumanML3D, Testing on KITML (the TMR Method Is Also Evaluated by Training Only on KITML: See "–" Option in the JDL Column)

| Protocol | Method | JDL | Motion-to-Text Retrieval | | | | | | Text-to-Motion Retrieval | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | MedR↓ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | MedR↓ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | Rsum↑ |
| (a) All | TMR [41] | – | 15.83 | 10.47 | 13.82 | 21.25 | 29.01 | 40.67 | 16.67 | 6.32 | 11.87 | 17.69 | 26.25 | 39.35 | 216.70 |
| | TMR [41] | ✓ | 11.33 | 13.78 | 18.07 | 26.68 | 36.43 | 49.15 | 10.33 | 9.37 | 18.24 | 25.15 | 35.37 | 49.79 | 282.03 |
| | Rmt* [64] | ✓ | 13.67 | 12.55 | 14.93 | 23.24 | 31.72 | 44.61 | 13.33 | 7.59 | 14.46 | 19.64 | 29.47 | 43.81 | 242.02 |
| | Rmt [64] | ✓ | 12.00 | 13.11 | 15.18 | 24.00 | 33.34 | 47.20 | 11.50 | 8.23 | 16.20 | 22.69 | 31.72 | 47.45 | 259.12 |
| | MoT [36] | ✓ | 11.83 | 12.25 | 15.99 | 25.32 | 34.10 | 48.26 | 10.67 | 8.99 | 17.60 | 24.17 | 33.50 | 48.90 | 269.08 |
| | MoT++ | ✓ | 11.50 | 14.42 | 17.94 | 26.63 | 35.54 | 48.60 | 10.83 | 8.99 | 17.81 | 24.39 | 34.73 | 49.83 | 278.88 |
| (b) All with threshold | TMR [41] | – | 9.50 | 18.91 | 23.58 | 31.76 | 40.16 | 51.61 | 6.00 | 22.43 | 29.94 | 38.17 | 47.37 | 59.67 | 363.60 |
| | TMR [41] | ✓ | 6.42 | 23.24 | 28.75 | 38.93 | 48.60 | 62.47 | 4.00 | 27.02 | 37.45 | 48.47 | 58.06 | 70.57 | 443.56 |
| | Rmt* [64] | ✓ | 7.50 | 22.82 | 26.68 | 36.98 | 45.21 | 58.27 | 5.00 | 22.90 | 33.72 | 43.60 | 53.52 | 67.73 | 411.43 |
| | Rmt [64] | ✓ | 7.42 | 22.78 | 26.08 | 37.32 | 46.18 | 62.08 | 4.00 | 26.46 | 36.47 | 46.69 | 56.32 | 71.42 | 431.80 |
| | MoT [36] | ✓ | 6.75 | 21.50 | 27.99 | 37.96 | 47.12 | 60.56 | 4.17 | 27.99 | 36.43 | 46.61 | 57.13 | 70.78 | 434.07 |
| | MoT++ | ✓ | 6.67 | 21.84 | 27.18 | 38.04 | 47.50 | 62.09 | 4.00 | 24.34 | 37.83 | 47.63 | 57.89 | 72.39 | 436.73 |
| (c) Dissim. subset | TMR [41] | – | 3.42 | 35.03 | 45.24 | 57.48 | 65.65 | 79.93 | 3.33 | 25.17 | 42.18 | 50.68 | 63.61 | 79.25 | 544.22 |
| | TMR [41] | ✓ | 2.42 | 44.22 | 54.76 | 68.37 | 75.85 | 85.71 | 2.67 | 28.57 | 48.64 | 59.52 | 73.47 | 86.05 | 625.16 |
| | Rmt* [64] | ✓ | 2.67 | 41.16 | 50.68 | 62.58 | 71.09 | 82.65 | 2.00 | 29.59 | 52.38 | 62.58 | 72.79 | 86.73 | 612.23 |
| | Rmt [64] | ✓ | 2.83 | 41.16 | 50.68 | 64.29 | 75.17 | 91.16 | 2.33 | 28.57 | 51.70 | 65.31 | 77.89 | 92.52 | 638.45 |
| | MoT [36] | ✓ | 2.58 | 41.50 | 53.74 | 65.65 | 76.19 | 86.05 | 2.67 | 28.91 | 49.66 | 61.56 | 79.25 | 91.16 | 633.67 |
| | MoT++ | ✓ | 2.08 | 48.30 | 58.16 | 68.03 | 77.55 | 88.10 | 2.33 | 28.23 | 50.34 | 60.88 | 77.21 | 90.14 | 646.94 |
| (d) Small batches | TMR [41] | – | 1.41 | 52.91 | 72.05 | 81.68 | 90.71 | 96.09 | 1.45 | 52.04 | 72.22 | 81.60 | 90.63 | 95.87 | 785.80 |
| | TMR [41] | ✓ | 1.16 | 59.29 | 80.34 | 87.46 | 93.53 | 96.83 | 1.15 | 59.42 | 79.86 | 87.50 | 94.01 | 97.18 | 835.42 |
| | Rmt* [64] | ✓ | 1.24 | 55.95 | 77.47 | 85.81 | 92.32 | 97.31 | 1.27 | 55.77 | 77.43 | 87.46 | 94.10 | 97.49 | 821.11 |
| | Rmt [64] | ✓ | 1.14 | 57.81 | 77.47 | 86.07 | 93.75 | 98.05 | 1.09 | 60.07 | 80.25 | 88.50 | 95.27 | 98.18 | 835.42 |
| | MoT [36] | ✓ | 1.23 | 57.12 | 77.12 | 86.03 | 92.88 | 96.92 | 1.10 | 60.76 | 80.51 | 88.71 | 94.88 | 98.05 | 832.98 |
| | MoT++ | ✓ | 1.18 | 59.99 | 79.73 | 87.20 | 93.71 | 97.44 | 1.12 | 59.24 | 80.04 | 88.58 | 95.57 | 98.35 | 839.85 |
| **Average over protocols** | TMR [41] | – | 7.54 | 29.33 | 38.67 | 48.04 | 56.38 | 67.08 | 6.86 | 26.49 | 39.05 | 47.03 | 56.96 | 68.54 | 477.57 |
| | TMR [41] | ✓ | **5.33** | 35.13 | 45.48 | **55.36** | **63.60** | 73.54 | **4.54** | 31.10 | 46.05 | 55.16 | 65.23 | 75.90 | 546.55 |
| | Rmt* [64] | ✓ | 6.27 | 33.12 | 42.44 | 52.15 | 60.08 | 70.71 | 5.40 | 28.96 | 44.50 | 53.32 | 62.47 | 73.94 | 521.69 |
| | Rmt [64] | ✓ | 5.85 | 33.71 | 42.35 | 52.92 | 62.11 | **74.62** | 4.73 | 30.83 | 46.16 | **55.80** | 65.30 | 77.39 | 541.19 |
| | MoT [36] | ✓ | 5.60 | 33.09 | 43.71 | 53.74 | 62.57 | 72.95 | 4.65 | **31.66** | 46.05 | 55.26 | 66.19 | 77.22 | 542.44 |
| | MoT++ | ✓ | 5.36 | **36.14** | **45.75** | 54.98 | 63.57 | 74.06 | 4.57 | 30.20 | **46.50** | 55.37 | **66.35** | **77.68** | **550.60** |

Bold font highlights the best achieved results.

### 4.2 Results of JDL and Cross-Dataset Learning

We first analyze the effect of JDL. Table 1 reports the results of selected methods evaluated on KITML as the test dataset. The JDL column indicates whether both KITML and HumanML3D datasets were used simultaneously for training ("✓" option), or just the KITML dataset was utilized ("–" option). When focusing on the last table section (i.e., "average over protocols"), we can observe the clearly best result (Rsum 550.60) of the proposed MoT++ method compared to the result of state-of-the-art TMR [41] and Rehamot method [64]. To be fair, we also evaluate the provided TMR,[3] MoT,[4] and Rehamot[5] implementations using the JDL approach—we can still observe that our MoT++ outperforms both these approaches on average (Rsum 550.60 vs. 546.55 and 541.19), noticeably achieving an average improvement of around 0.7% in R@10 on motion-to-text and 1.7% on text-to-motion retrieval scenarios with respect TMR, only obtaining slightly worse results on "All" and "All with threshold" protocols. A similar trend can be observed in Table 2 where HumanML3D is used as the test dataset. Again, the proposed MoT++ approach outperforms existing approaches, with an increase on the R@10 metric of 1.2% on motion-to-text and of 1.0% on text-to-motion retrieval, also obtaining the best median rank (MedR) values. Interestingly, JDL seems not to help TMR trained purely on HumanML3D since KITML is quite a small and motion-specific dataset that provides limited generalizability for HumanML3D. Noticeably, our method seems to always outperform the recent Rehamot architecture when employing cosine similarity (Rmt*).

[3] https://github.com/Mathux/TMR.
[4] https://github.com/mesnico/text-to-motion-retrieval.
[5] https://github.com/eanson023/rehamot.

Table 2. JDL: Training on KITML+HumanML3D, Testing on HumanML3D (the TMR Method Is Also Evaluated by Training Only on HumanML3D: See "–" Option in the JDL Column)

| Protocol | Method | JDL | Motion-to-Text Retrieval | | | | | | Text-to-Motion Retrieval | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | MedR↓ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | MedR↓ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | Rsum↑ |
| (a) All | TMR [41] | – | 28.17 | 8.72 | 11.07 | 15.85 | 21.35 | 31.20 | 27.67 | 5.35 | 10.11 | 13.45 | 19.33 | 30.74 | 167.17 |
| | TMR [41] | ✓ | 27.33 | 8.75 | 10.96 | 16.40 | 21.93 | 31.61 | 27.67 | 5.48 | 10.17 | 13.53 | 19.26 | 30.40 | 168.49 |
| | Rmt* [64] | ✓ | 37.33 | 6.21 | 8.22 | 12.55 | 17.24 | 25.56 | 36.00 | 3.79 | 7.54 | 10.65 | 15.61 | 25.10 | 132.47 |
| | Rmt [64] | ✓ | 34.25 | 6.69 | 8.44 | 12.73 | 17.71 | 25.86 | 30.67 | 4.30 | 8.12 | 11.17 | 16.65 | 26.68 | 138.35 |
| | MoT [36] | ✓ | 35.83 | 6.41 | 8.26 | 12.61 | 17.24 | 26.23 | 34.67 | 4.27 | 7.92 | 10.83 | 16.03 | 25.91 | 135.71 |
| | MoT++ | ✓ | 25.92 | 8.79 | 11.45 | 17.00 | 22.67 | 32.22 | 26.33 | 5.24 | 10.00 | 14.07 | 20.28 | 31.01 | 172.73 |
| (b) All with threshold | TMR [41] | – | 22.00 | 12.23 | 14.39 | 20.19 | 26.11 | 36.26 | 18.33 | 11.42 | 15.67 | 21.19 | 27.71 | 39.47 | 224.64 |
| | TMR [41] | ✓ | 21.00 | 12.24 | 14.43 | 20.75 | 26.77 | 36.93 | 17.33 | 11.38 | 15.73 | 21.04 | 28.03 | 39.85 | 227.15 |
| | Rmt* [64] | ✓ | 30.17 | 10.76 | 12.66 | 17.35 | 21.97 | 30.09 | 20.67 | 9.92 | 14.56 | 19.51 | 25.71 | 36.51 | 199.04 |
| | Rmt [64] | ✓ | 28.50 | 11.34 | 12.69 | 17.85 | 22.67 | 30.89 | 19.00 | 10.36 | 14.34 | 19.71 | 26.85 | 38.03 | 204.73 |
| | MoT [36] | ✓ | 28.33 | 10.08 | 12.20 | 17.49 | 22.73 | 31.87 | 21.67 | 9.36 | 13.53 | 18.29 | 24.88 | 36.25 | 196.68 |
| | MoT++ | ✓ | 20.75 | 11.57 | 14.09 | 20.47 | 26.67 | 36.73 | 17.00 | 10.83 | 15.38 | 21.28 | 29.03 | 40.58 | 226.63 |
| (c) Dissim. subset | TMR [41] | – | 1.83 | 49.33 | 67.00 | 73.33 | 81.00 | 88.67 | 2.00 | 47.00 | 67.00 | 72.33 | 80.00 | 88.00 | 713.66 |
| | TMR [41] | ✓ | 2.00 | 46.33 | 67.67 | 73.67 | 83.00 | 88.33 | 2.00 | 45.67 | 65.67 | 74.00 | 82.00 | 88.67 | 715.01 |
| | Rmt* [64] | ✓ | 2.00 | 47.00 | 61.33 | 67.67 | 77.33 | 86.33 | 2.00 | 45.67 | 61.33 | 67.67 | 77.00 | 85.67 | 677.00 |
| | Rmt [64] | ✓ | 2.00 | 41.33 | 59.33 | 69.67 | 78.33 | 86.67 | 2.00 | 43.00 | 62.33 | 70.67 | 81.00 | 87.33 | 679.66 |
| | MoT [36] | ✓ | 2.00 | 42.67 | 61.00 | 71.67 | 80.00 | 89.33 | 2.00 | 42.33 | 61.33 | 72.33 | 82.33 | 90.00 | 692.99 |
| | MoT++ | ✓ | 1.83 | 48.33 | 64.33 | 74.00 | 82.33 | 90.33 | 1.83 | 49.00 | 66.67 | 72.33 | 80.33 | 89.00 | 716.65 |
| (d) Small batches | TMR [41] | – | 1.02 | 68.03 | 82.38 | 87.86 | 92.32 | 96.27 | 1.01 | 67.32 | 82.00 | 87.43 | 92.22 | 96.36 | 852.19 |
| | TMR [41] | ✓ | 1.02 | 68.45 | 82.89 | 88.26 | 92.74 | 96.47 | 1.02 | 68.45 | 82.71 | 87.94 | 92.52 | 96.40 | 856.83 |
| | Rmt* [64] | ✓ | 1.05 | 63.57 | 78.86 | 85.76 | 91.59 | 96.74 | 1.05 | 63.93 | 78.98 | 85.44 | 91.19 | 96.55 | 832.61 |
| | Rmt [64] | ✓ | 1.04 | 65.25 | 81.01 | 87.45 | 92.72 | 96.88 | 1.03 | 66.10 | 81.56 | 87.51 | 92.52 | 96.88 | 847.88 |
| | MoT [36] | ✓ | 1.04 | 64.83 | 80.47 | 86.69 | 92.35 | 96.83 | 1.04 | 64.99 | 80.51 | 86.91 | 91.91 | 96.59 | 842.08 |
| | MoT++ | ✓ | 1.01 | 69.02 | 83.71 | 89.27 | 93.71 | 97.35 | 1.01 | 68.43 | 82.74 | 88.69 | 93.41 | 97.31 | 863.64 |
| **Average over protocols** | TMR [41] | – | 13.26 | **34.58** | 43.71 | 49.31 | 55.19 | 63.10 | 12.25 | 32.77 | 43.69 | 48.60 | 54.82 | 63.64 | 489.41 |
| | TMR [41] | ✓ | 12.84 | 33.94 | **43.99** | 49.77 | 56.11 | 63.34 | 12.00 | 32.74 | 43.57 | **49.13** | 55.45 | 63.83 | 491.87 |
| | Rmt* [64] | ✓ | 17.64 | 31.88 | 40.27 | 45.83 | 52.03 | 59.68 | 14.93 | 30.83 | 40.60 | 45.82 | 52.38 | 60.96 | 460.28 |
| | Rmt [64] | ✓ | 16.45 | 31.15 | 40.37 | 46.92 | 52.86 | 60.07 | 13.18 | 30.94 | 41.59 | 47.27 | 54.26 | 62.23 | 467.66 |
| | MoT [36] | ✓ | 16.80 | 31.00 | 40.48 | 47.12 | 53.08 | 61.06 | 14.84 | 30.24 | 40.83 | 47.09 | 53.79 | 62.19 | 466.88 |
| | MoT++ | ✓ | **12.38** | 34.43 | 43.40 | **50.18** | **56.35** | **64.16** | **11.54** | **33.38** | **43.70** | 49.09 | **55.76** | **64.47** | **494.92** |

Bold font highlights the best achieved results.

This is the Rehamot version most similar to our setup where motion and text representations are compared using cosine similarity instead of the more complex CPS [64].

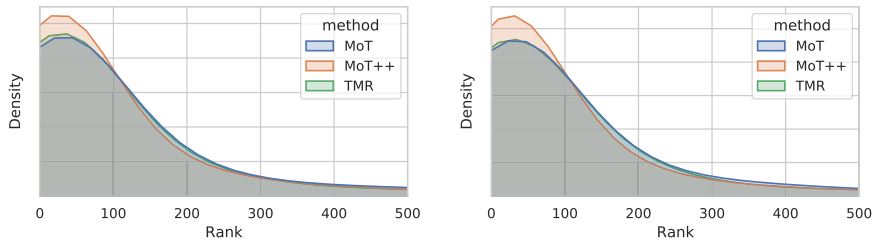
To also have a better understanding of the generalization abilities of the proposed models and the explored baselines, we also evaluate the *cross-dataset* scenario by training on HumanML3D and testing on KITML (the opposite variant, i.e., training on KITML and testing on HumanML3D, is less meaningful due to the much richer nature of HumanML3D). The results in Table 3 demonstrate that MoT++ again reaches the best result on average (Rsum 527.4) in comparison with the results of TMR, MoT (Rsum ~516.5), and Rehamot (Rsum 463.13), with an increase on R@10 metric of 1.5% on motion-to-text and of 2.0% on text-to-motion with respect to MoT, which noticeably obtains better results with respect to TMR. Particularly, we can observe the difficulty of Rehamot in this generalization scenario. These results suggest that the two MoT-based motion encoders are more prone to generalize across diverse datasets. Another interesting observation is that the result of MoT++ (Rsum 527.4) is much better than the result of TMR trained purposely on KITML (Rsum 477.6 in Table 1), which proves the ability of MoT++ to generalize well on unseen data.

To further show the positive effect of CCCL and joint-dataset training on the retrieval results, we plot the distribution of the ranks—positions of the first relevant objects retrieved—in Figure 4 for all the test queries, on HumanML3D. We can observe that MoT++ has clearly more queries answered with a better quality—a query match is located within the ~50 nearest neighbors in roughly about 15–20% more queries. The high peak is due to the normalization computed on the whole distribution, although the figure shows only the top ranks. The higher peak signifies

Table 3. Cross-Dataset Inference: Training on HumanML3D, Testing on KITML

| Protocol | Method | Motion-to-Text Retrieval | | | | | | Text-to-Motion Retrieval | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MedR↓ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | MedR↓ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | Rsum↑ |
| (a) All | TMR [41] | 12.33 | 12.80 | 15.31 | 24.21 | 33.25 | 46.48 | 13.33 | 8.10 | 15.65 | 21.88 | 31.00 | 44.87 | 253.55 |
| | Rmt* [64] | 37.92 | 6.43 | 7.99 | 12.42 | 17.27 | 25.41 | 38.00 | 3.98 | 7.81 | 10.55 | 15.34 | 25.26 | 132.46 |
| | Rmt [64] | 34.50 | 6.52 | 8.29 | 12.45 | 17.48 | 25.59 | 31.42 | 4.15 | 7.90 | 10.96 | 16.26 | 26.85 | 136.45 |
| | MoT [36] | 13.42 | 12.38 | 15.40 | 22.65 | 30.57 | 45.21 | 15.17 | 7.12 | 14.21 | 19.59 | 28.37 | 41.52 | 237.02 |
| | MoT++ | 12.00 | 12.47 | 15.95 | 24.13 | 33.50 | 46.95 | 12.83 | 7.17 | 14.38 | 20.44 | 29.94 | 46.02 | 250.95 |
| (b) All with threshold | TMR [41] | 7.33 | 22.94 | 26.63 | 38.34 | 46.10 | 59.12 | 5.00 | 25.83 | 33.59 | 42.45 | 51.65 | 64.46 | 411.11 |
| | Rmt* [64] | 31.83 | 10.83 | 12.13 | 17.48 | 22.19 | 30.14 | 22.00 | 9.64 | 14.57 | 18.90 | 25.53 | 36.68 | 198.09 |
| | Rmt [64] | 29.42 | 11.08 | 12.39 | 17.51 | 22.62 | 30.49 | 19.00 | 9.79 | 13.83 | 18.71 | 25.38 | 38.17 | 199.97 |
| | MoT [36] | 7.83 | 21.50 | 26.25 | 35.03 | 43.00 | 56.53 | 5.33 | 21.97 | 32.06 | 38.76 | 49.70 | 63.99 | 388.79 |
| | MoT++ | 7.00 | 20.99 | 26.76 | 36.77 | 47.12 | 59.84 | 5.00 | 22.86 | 32.19 | 43.34 | 53.01 | 67.35 | 410.23 |
| (c) Dissim. subset | TMR [41] | 3.00 | 36.05 | 47.96 | 64.63 | 73.13 | 83.33 | 3.00 | 26.19 | 44.56 | 57.14 | 69.05 | 82.65 | 584.69 |
| | Rmt* [64] | 2.00 | 43.33 | 57.67 | 63.33 | 75.00 | 86.67 | 2.00 | 39.00 | 56.33 | 66.00 | 77.33 | 88.00 | 652.66 |
| | Rmt [64] | 2.00 | 40.33 | 58.33 | 68.67 | 77.33 | 85.67 | 2.00 | 43.67 | 61.67 | 70.67 | 76.67 | 86.67 | 669.68 |
| | MoT [36] | 2.67 | 39.80 | 50.68 | 64.63 | 78.57 | 88.09 | 2.17 | 32.65 | 52.38 | 65.31 | 78.23 | 89.80 | 640.14 |
| | MoT++ | 2.58 | 41.84 | 53.06 | 66.67 | 77.21 | 88.10 | 2.67 | 27.89 | 49.66 | 61.90 | 74.15 | 88.78 | 629.26 |
| (d) Small batches | TMR [41] | 1.25 | 56.77 | 75.74 | 85.24 | 93.36 | 97.88 | 1.26 | 55.43 | 75.52 | 85.98 | 93.32 | 97.70 | 816.94 |
| | Rmt* [64] | 1.06 | 63.02 | 78.50 | 85.19 | 91.10 | 96.65 | 1.08 | 62.46 | 78.03 | 85.02 | 90.97 | 96.49 | 827.43 |
| | Rmt [64] | 1.04 | 64.89 | 81.02 | 87.49 | 92.29 | 96.99 | 1.03 | 66.00 | 81.46 | 87.20 | 92.29 | 96.82 | 846.45 |
| | MoT [36] | 1.31 | 54.30 | 73.87 | 82.64 | 92.67 | 98.01 | 1.40 | 52.26 | 72.74 | 82.59 | 92.80 | 98.27 | 800.15 |
| | MoT++ | 1.19 | 58.51 | 77.69 | 86.15 | 93.19 | 97.53 | 1.24 | 56.03 | 75.09 | 84.29 | 93.06 | 97.70 | 819.24 |
| **Average over protocols** | TMR [41] | 5.98 | 32.14 | 41.41 | 53.10 | 61.46 | 71.70 | 5.65 | 28.89 | 42.33 | 51.86 | 61.26 | 72.42 | 516.57 |
| | Rmt* [64] | 18.20 | 30.90 | 39.07 | 44.61 | 51.39 | 59.72 | 15.77 | 28.77 | 39.18 | 45.12 | 52.30 | 61.61 | 452.67 |
| | Rmt [64] | 16.74 | 30.70 | 40.01 | 46.53 | 52.43 | 59.68 | 13.36 | **30.90** | 41.21 | 46.89 | 52.65 | 62.13 | 463.13 |
| | MoT [36] | 6.31 | 32.00 | 41.55 | 51.24 | 61.20 | 71.96 | 6.02 | 28.50 | **42.85** | 51.56 | 62.28 | 73.40 | 516.54 |
| | MoT++ | **5.69** | **33.45** | **43.36** | **53.43** | **62.75** | **73.10** | **5.44** | 28.49 | 42.83 | **52.49** | **62.54** | **74.96** | **527.40** |

Bold font highlights the best achieved results.




(a) Text-to-motion rank distribution (b) Motion-to-text rank distribution

Fig. 4. Distributions of ranks ($x$-axis) of relevant objects retrieved for (a) text-to-motion and (b) motion-to-text scenarios using JDL: training on KITML+HumanML3D, testing on HumanML3D.

that our method is more able to handle the distribution queue, avoiding large outliers and bringing them closer to the top.

Concerning training and inference times, MoT, MoT++, Rmt*, and TMR are very much comparable. On a single Nvidia-A100 GPU, training takes in the order of 1.5–2.0 minutes for a single epoch, while inference takes 10–12 seconds on HumanML and 2 seconds on KITML (considering both the feature extraction and search times). These very low latency times are due to the underlying two-stage approach, in which text and motion features can be pre-computed and efficiently matched with a simple cosine similarity.

In summary, JDL enhances the performance of all methods across various retrieval tasks and protocols, while also keeping a high efficiency. MoT++ with JDL generally provides the best performance, making it a robust choice for text-motion retrieval tasks.

Table 4. CCCL Ablations on MoT++ Using Single-Dataset Learning (Training on HumanML3D: "–" Option in JDL) and JDL (Training on KITML+HumanML3D: "✓" Option in JDL), Testing on HumanML3D

| Protocol | Loss function | JDL | Motion-to-Text Retrieval | | | | | | Text-to-Motion Retrieval | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | MedR↓ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | MedR↓ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | Rsum↑ |
| (a) All | InfoNCE+F [41] | – | 29.00 | 8.32 | 10.40 | 15.79 | 21.22 | 30.03 | 30.00 | 4.94 | 9.50 | 12.92 | 18.62 | 29.46 | 161.20 |
| | InfoNCE+F [41] | ✓ | 30.00 | 8.58 | 11.08 | 15.62 | 20.86 | 30.47 | 30.33 | 4.77 | 9.47 | 12.64 | 18.42 | 28.74 | 160.65 |
| | CCCL | – | 26.50 | 9.06 | 11.54 | 16.49 | 22.08 | 31.70 | 27.17 | 5.55 | 10.67 | 14.10 | 20.31 | 31.21 | 172.71 |
| | CCCL | ✓ | 25.92 | 8.79 | 11.45 | 17.00 | 22.67 | 32.22 | 26.33 | 5.24 | 10.00 | 14.07 | 20.28 | 31.01 | 172.73 |
| (b) All with threshold | InfoNCE+F [41] | – | 23.00 | 12.00 | 13.77 | 20.04 | 25.92 | 35.08 | 19.33 | 11.14 | 15.00 | 20.50 | 26.96 | 38.18 | 218.59 |
| | InfoNCE+F [41] | ✓ | 23.00 | 11.92 | 14.26 | 19.71 | 25.52 | 35.64 | 20.00 | 10.52 | 14.53 | 19.96 | 26.47 | 37.77 | 216.30 |
| | CCCL | – | 21.33 | 11.89 | 14.00 | 20.02 | 26.19 | 36.24 | 17.67 | 11.24 | 16.05 | 21.45 | 28.52 | 40.05 | 225.65 |
| | CCCL | ✓ | 20.75 | 11.57 | 14.09 | 20.47 | 26.67 | 36.73 | 17.00 | 10.83 | 15.38 | 21.28 | 29.03 | 40.58 | 226.63 |
| (c) Dissim. subset | InfoNCE+F [41] | – | 1.33 | 50.00 | 66.33 | 73.33 | 81.33 | 88.00 | 1.83 | 48.00 | 67.33 | 74.33 | 80.33 | 88.33 | 717.31 |
| | InfoNCE+F [41] | ✓ | 2.00 | 43.00 | 66.00 | 74.00 | 83.00 | 88.00 | 1.83 | 46.00 | 65.33 | 73.33 | 79.33 | 88.33 | 706.32 |
| | CCCL | – | 2.00 | 47.00 | 65.00 | 73.67 | 80.33 | 88.33 | 1.83 | 47.33 | 65.00 | 73.67 | 79.67 | 87.33 | 707.33 |
| | CCCL | ✓ | 1.83 | 48.33 | 64.33 | 74.00 | 82.33 | 90.33 | 1.83 | 49.00 | 66.67 | 72.33 | 80.33 | 89.00 | 716.65 |
| (d) Small batches | InfoNCE+F [41] | – | 1.04 | 67.37 | 81.55 | 87.29 | 91.75 | 95.87 | 1.03 | 66.68 | 81.21 | 86.80 | 91.70 | 96.16 | 846.38 |
| | InfoNCE+F [41] | ✓ | 1.02 | 67.72 | 82.28 | 87.74 | 92.04 | 95.96 | 1.02 | 66.68 | 81.65 | 87.21 | 91.79 | 95.97 | 849.04 |
| | CCCL | – | 1.01 | 68.74 | 82.82 | 88.22 | 93.01 | 97.14 | 1.01 | 68.13 | 82.38 | 87.60 | 92.37 | 97.06 | 857.47 |
| | CCCL | ✓ | 1.01 | 69.02 | 83.71 | 89.27 | 93.71 | 97.35 | 1.01 | 68.43 | 82.74 | 88.69 | 93.41 | 97.31 | 863.64 |
| **Average over protocols** | InfoNCE+F [41] | – | 13.59 | 34.42 | 43.01 | 49.11 | 55.06 | 62.25 | 13.05 | 32.69 | 43.26 | 48.64 | 54.41 | 63.04 | 485.89 |
| | InfoNCE+F [41] | ✓ | 14.00 | 32.80 | 43.40 | 49.26 | 55.35 | 62.52 | 13.30 | 31.99 | 42.74 | 48.28 | 54.00 | 62.70 | 483.04 |
| | CCCL | – | 12.71 | 34.17 | 43.34 | 49.60 | 55.40 | 63.35 | 11.92 | 33.06 | 43.53 | **49.20** | 55.22 | 63.91 | 490.78 |
| | CCCL | ✓ | **12.38** | **34.43** | **43.40** | **50.18** | **56.35** | **64.16** | **11.54** | **33.38** | **43.70** | 49.09 | **55.76** | **64.47** | **494.92** |

Bold font highlights the best achieved results.

### 4.3 Ablation Study

In this section, we inquire about the importance of all the introduced components. Specifically, (i) we study if JDL, together with the proposed CCCL loss, effectively contributes to improving the overall performance in both text-to-motion and motion-to-text scenarios; (ii) we understand the role of the $\lambda$ swipe hyper-parameters introduced in CCCL; (iii) we provide a quantitative evaluation of motion-to-motion retrieval—a nice by-product of CCCL; and (iv) we provide some qualitative results by comparing our outcomes with the state-of-the-art TMR method.

*4.3.1 Role of CCCL + JDL.* We aim to explore the important roles that both the JDL and our proposed CCCL objective have on the final model. In Table 4, we report the results on MoT++ by alternatively enabling or disabling the JDL and using either InfoNCE with filtering (InfoNCE+F) or CCCL. As we can notice, the best overall results are obtained with the combination of joint-dataset training and CCCL, obtaining an overall increase in Rsum between 485.89 and 494.92, with an increase of 3.0% on R@10 on motion-to-text and 2.2% on text-to-motion. These results demonstrate the nice effect that the availability of more data and the careful constraints imposed by CCCL on the common space have on the overall model.

*4.3.2 CCCL Hyper-Parameters.* Our loss formulation is driven by two main hyper-parameters, $t_{\text{start}}$ and $t_{\text{end}}$, which define the start and end epochs of the linear transition between the text supervision enforced by the teacher model $\mathcal{E}_t^{\text{ref}}$ and the self-sustained regime where the teacher is fully disabled. In Table 5, we report experiments performed on MoT++, on the JDL setup, for different ranges of $t_{\text{start}}$ and $t_{\text{end}}$, where we indicate as "CCCL x-y" the experiment performed using $t_{\text{start}} = x$ and $t_{\text{end}} = y$. We also report the edge cases in which the text teacher is always disabled (CCCL self) and in which it is instead always active (CCCL supervised). As we can notice, the best result is obtained when $\lambda$ is varied between epochs 40 and 100 (which is the configuration employed in Tables 1, 2, and 3). By looking at the final Rsum value over the averaged measures, it is interesting to notice that almost all the configurations (comprising the fully self-sustained one) can surpass the InfoNCE with filtering (InfoNCE+F) employed by TMR in [41]. Notably,

Table 5. CCCL Ablations on MoT++ Using JDL: Training on KITML+HumanML3D, Testing on HumanML3D

| Protocol | Loss function | Motion-to-Text Retrieval | | | | | | Text-to-Motion Retrieval | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MedR↓ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | MedR↓ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | Rsum↑ |
| (a) All | InfoNCE+F [41] | 30.00 | 8.58 | 11.08 | 15.62 | 20.86 | 30.47 | 30.33 | 4.77 | 9.47 | 12.64 | 18.42 | 28.74 | 160.65 |
| | CCCL self | 27.67 | 8.46 | 10.93 | 16.13 | 21.79 | 31.84 | 28.00 | 5.26 | 9.84 | 13.25 | 19.44 | 30.06 | 167.00 |
| | CCCL 40-100 | 25.92 | 8.79 | 11.45 | 17.00 | 22.67 | 32.22 | 26.33 | 5.24 | 10.00 | 14.07 | 20.28 | 31.01 | 172.73 |
| | CCCL 80-140 | 27.00 | 8.86 | 11.19 | 16.58 | 22.57 | 32.29 | 27.33 | 5.12 | 9.93 | 13.50 | 19.67 | 31.21 | 170.92 |
| | CCCL 140-200 | 28.17 | 9.15 | 11.60 | 16.61 | 21.84 | 31.27 | 28.33 | 5.33 | 10.35 | 13.96 | 19.94 | 29.99 | 170.04 |
| | CCCL supervised | 30.50 | 8.45 | 10.85 | 15.77 | 20.99 | 29.88 | 30.67 | 4.76 | 9.39 | 12.80 | 18.77 | 29.53 | 161.19 |
| (b) All with threshold | InfoNCE+F [41] | 23.00 | 11.92 | 14.26 | 19.71 | 25.52 | 35.64 | 20.00 | 10.52 | 14.53 | 19.96 | 26.47 | 37.77 | 216.30 |
| | CCCL self | 21.58 | 11.34 | 13.53 | 20.00 | 26.00 | 36.61 | 18.00 | 11.06 | 15.25 | 20.56 | 27.76 | 39.33 | 221.44 |
| | CCCL 40-100 | 20.75 | 11.57 | 14.09 | 20.47 | 26.67 | 36.73 | 17.00 | 10.83 | 15.38 | 21.28 | 29.03 | 40.58 | 226.63 |
| | CCCL 80-140 | 21.50 | 11.65 | 13.92 | 20.50 | 26.89 | 36.79 | 17.00 | 10.52 | 15.27 | 20.81 | 28.01 | 40.59 | 224.95 |
| | CCCL 140-200 | 21.67 | 12.11 | 14.39 | 20.20 | 26.09 | 36.07 | 17.67 | 11.31 | 15.74 | 21.07 | 28.11 | 38.92 | 224.01 |
| | CCCL supervised | 23.33 | 11.52 | 13.85 | 19.53 | 25.47 | 34.89 | 19.33 | 10.19 | 14.85 | 20.01 | 27.15 | 38.38 | 215.84 |
| (c) Dissim. subset | InfoNCE+F [41] | 2.00 | 43.00 | 66.00 | 74.00 | 83.00 | 88.00 | 1.83 | 46.00 | 65.33 | 73.33 | 79.33 | 88.33 | 706.32 |
| | CCCL self | 1.67 | 48.00 | 66.33 | 74.67 | 83.33 | 90.67 | 1.67 | 47.33 | 65.33 | 73.67 | 82.67 | 90.33 | 722.33 |
| | CCCL 40-100 | 1.83 | 48.33 | 64.33 | 74.00 | 82.33 | 90.33 | 1.83 | 49.00 | 66.67 | 72.33 | 80.33 | 89.00 | 716.65 |
| | CCCL 80-140 | 2.00 | 45.67 | 62.33 | 70.33 | 80.00 | 88.67 | 1.83 | 47.33 | 63.33 | 69.67 | 76.33 | 88.33 | 691.99 |
| | CCCL 140-200 | 1.83 | 48.00 | 62.33 | 72.33 | 81.33 | 89.00 | 1.67 | 48.67 | 63.67 | 69.67 | 79.67 | 86.67 | 701.34 |
| | CCCL supervised | 1.67 | 47.67 | 63.33 | 72.67 | 79.67 | 87.33 | 2.00 | 45.67 | 65.00 | 71.33 | 79.00 | 86.67 | 698.34 |
| (d) Small batches | InfoNCE+F [41] | 1.02 | 67.72 | 82.28 | 87.74 | 92.04 | 95.96 | 1.02 | 66.68 | 81.65 | 87.21 | 91.79 | 95.97 | 849.04 |
| | CCCL self | 1.00 | 68.51 | 83.12 | 88.69 | 93.57 | 97.39 | 1.02 | 67.66 | 82.24 | 88.17 | 93.34 | 97.31 | 860.00 |
| | CCCL 40-100 | 1.01 | 69.02 | 83.71 | 89.27 | 93.71 | 97.35 | 1.01 | 68.43 | 82.74 | 88.69 | 93.41 | 97.31 | 863.64 |
| | CCCL 80-140 | 1.01 | 68.21 | 82.88 | 88.69 | 93.41 | 97.36 | 1.02 | 67.81 | 81.93 | 87.92 | 93.01 | 97.21 | 858.43 |
| | CCCL 140-200 | 1.01 | 67.57 | 81.99 | 87.70 | 93.24 | 97.27 | 1.01 | 67.01 | 81.45 | 87.17 | 92.70 | 97.16 | 853.26 |
| | CCCL supervised | 1.02 | 66.04 | 80.67 | 86.52 | 91.99 | 96.67 | 1.03 | 65.78 | 79.65 | 85.71 | 91.52 | 96.38 | 840.93 |
| **Average over protocols** | InfoNCE+F [41] | 14.00 | 32.80 | 43.40 | 49.26 | 55.35 | 62.52 | 13.30 | 31.99 | 42.74 | 48.28 | 54.00 | 62.70 | 483.04 |
| | CCCL self | 12.98 | 34.08 | **43.48** | 49.87 | 56.17 | 64.13 | 12.17 | 32.83 | 43.17 | 48.91 | **55.80** | 64.26 | 492.70 |
| | CCCL 40-100 | **12.38** | **34.43** | 43.40 | **50.18** | **56.35** | **64.16** | **11.54** | **33.38** | **43.70** | **49.09** | 55.76 | **64.47** | **494.92** |
| | CCCL 80-140 | 12.88 | 33.60 | 42.58 | 49.03 | 55.72 | 63.78 | 11.80 | 32.70 | 42.61 | 47.97 | 54.26 | 64.34 | 486.59 |
| | CCCL 140-200 | 13.17 | 34.21 | 42.58 | 49.21 | 55.63 | 63.40 | 12.17 | 33.08 | 42.80 | 47.96 | 55.10 | 63.18 | 487.15 |
| | CCCL supervised | 14.13 | 33.42 | 42.18 | 48.62 | 54.53 | 62.20 | 13.26 | 31.60 | 42.22 | 47.46 | 54.11 | 62.74 | 479.08 |

We indicate as "CCCL x-y" the experiment performed using $t_{\text{start}} = x$ and $t_{\text{end}} = y$. Bold font highlights the best achieved results.

the fully supervised scenario is the one achieving the worst results. This suggests that, while the supervision signal is helpful in the early training epochs, it is detrimental if kept active until the last iterations. This may be due to the fact that the text-motion domain is much more specific than the general-purpose knowledge conserved within the teacher $\mathcal{E}_t^{\text{ref}}$. Therefore, at a certain point, the text teacher stops providing sufficiently good supervision signals, introducing domain-specific noise that degrades the overall performance. Notice also how well the self-sustained method (CCCL self) works with respect to the InfoNCE even without the initial help of the teacher, further proving that this self-distillation of scores helps stabilize the whole network.

*4.3.3 MoT++ Architecture Variations.* In order to validate the effectiveness of the proposed MoT++, we perform some ablation studies on the specific spatial-temporal processing being used in the motion encoder's novel formulation, comparing it with the original MoT encoder. In particular, we report the MoT++ encoder using factorized encoder, divided-space-time, and joint-space-time configurations, together with the MoT configuration used in [36]. We kept all the other modules fixed to the main configuration employed in the article (i.e., ACTORStyle text encoder and the CCL 40-100 loss function). We run the experiments on the JDL configuration and test on HumanML3D. We report the results using the *All* protocol—the most representative protocol that searches the whole database—in Table 6.

*4.3.4 Evaluation on the Motion-X Dataset.* Motion-X is a recent large-scale 3D expressive whole-body human motion dataset. It includes 15.6 million precise 3D whole-body pose annotations and 81.1K motion sequences collected from a variety of scenes. It contains the HumanML3D dataset as a subset. Given that it includes both motions and textual descriptions from HumanML3D, we

Table 6. Ablations on MoT++ Using JDL: Training on KITML+HumanML3D, Testing on KITML

| Method | Motion-to-Text Retrieval | | | | | | Text-to-Motion Retrieval | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MedR↓ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | MedR↓ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | Rsum↑ |
| MoT [36] | 31.50 | 7.75 | 9.86 | 14.41 | 19.66 | 28.49 | 32.00 | 4.32 | 8.38 | 11.56 | 17.21 | 28.25 | 149.89 |
| MoT++ divided ST | 27.08 | **8.91** | 11.11 | 16.00 | 22.08 | 32.04 | 28.00 | **5.26** | **10.05** | 13.89 | 19.98 | 30.94 | 170.26 |
| MoT++ joint ST | 27.67 | 8.40 | 10.93 | 16.04 | 21.55 | 31.39 | 27.00 | 4.88 | 9.54 | 13.12 | 19.41 | 30.72 | 165.98 |
| MoT++ | **25.92** | 8.79 | **11.45** | **17.00** | **22.67** | **32.22** | **26.33** | 5.24 | 10.00 | **14.07** | **20.28** | **31.01** | **172.73** |

"ST" stands for space-time. Results reported on the *All* protocol. Bold font highlights the best achieved results.

Table 7. JDL: Training on Motion-X+HumanML3D, Testing on HumanML3D

| Method | Motion-to-Text Retrieval | | | | | | Text-to-Motion Retrieval | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MedR↓ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | MedR↓ | R@1↑ | R@2↑ | R@3↑ | R@5↑ | R@10↑ | Rsum↑ |
| TMR [41] | 30.67 | 7.79 | 9.96 | 14.94 | 20.14 | 29.20 | 30.33 | 4.63 | 8.97 | 12.37 | 18.15 | 28.57 | 154.72 |
| MoT [36] | 43.83 | 5.71 | 7.38 | 10.74 | 15.15 | 23.02 | 40.33 | 3.45 | 6.56 | 9.29 | 14.00 | 23.10 | 118.40 |
| MoT++ | **28.33** | **7.83** | **10.19** | **15.21** | **20.76** | **30.46** | **28.67** | **5.16** | **9.67** | **13.21** | **19.08** | **29.69** | **161.26** |

Results reported on the *All* protocol. Bold font highlights the best achieved results.

Table 8. Motion-to-Motion Retrieval Using JDL: Training on KITML+HumanML3D, Testing on HumanML3D

| Method | All | | Dissim. Subset | | Small Batches | | Average | |
|---|---|---|---|---|---|---|---|---|
| | mAP↑ | nDCG↑ | mAP↑ | nDCG↑ | mAP↑ | nDCG↑ | mAP↑ | nDCG↑ |
| TMR [41] | 0.728 | **0.906** | 0.850 | **0.922** | 0.757 | 0.867 | 0.778 | 0.898 |
| MoT [36] | 0.704 | 0.899 | 0.817 | 0.905 | 0.723 | 0.850 | 0.748 | 0.885 |
| MoT++ | 0.725 | 0.903 | 0.853 | **0.922** | 0.763 | 0.863 | 0.780 | 0.896 |
| MoT++ self | **0.734** | 0.905 | 0.845 | 0.918 | **0.777** | **0.877** | **0.785** | **0.900** |

mAP, mean Average Precision; nDCG, normalized Discounted Cumulative Gain. Bold font highlights the best achieved results.

removed the HumanML3D split from it to avoid training-testing interference. Despite being never used for text-to-motion or motion-to-text retrieval tasks, Motion-X has great potential in these scenarios due to its high expressiveness and variability. We train the different methods on Motion-X+HumanML3D and evaluate them on HumanML3D, reporting the *All* protocol's results in Table 7. We can notice that our method reaches the best results and outperforms the two previous methods on all metrics concerning text-to-motion retrieval, showing promising results also on this new challenging dataset.

*4.3.5 Motion-to-Motion Retrieval Evaluation.* One nice by-product of the proposed CCCL objective is the creation of a better uni-modal common space, which better organizes the motion features. This induced property enables the same proposed MoT++ motion encoder to be employed in the *query-by-example* setup, which consists of effectively finding motions mostly resembling a given query motion example, as discussed in many previous works [21, 32, 45, 46].

To quantitatively assess the ability of our model to perform motion-to-motion retrieval, we manually labeled the test set of the KITML dataset using a total of 90 different motion labels, which cluster the whole variety of samples present in KITML—ranging from common classes *walking* or *jumping* to more particular like *playingGuitar* or *kneelingDown*. We then evaluated the motion-to-motion retrieval effectiveness employing classical binary relevance metrics, specifically mean Average Precision [75] and normalized Discounted Cumulative Gain [19] with binary relevances.

The results are depicted in Table 8. Notice that we did not report the results for "All with threshold" protocol, as in this scenario, it would be particularly biased—as all the motions having

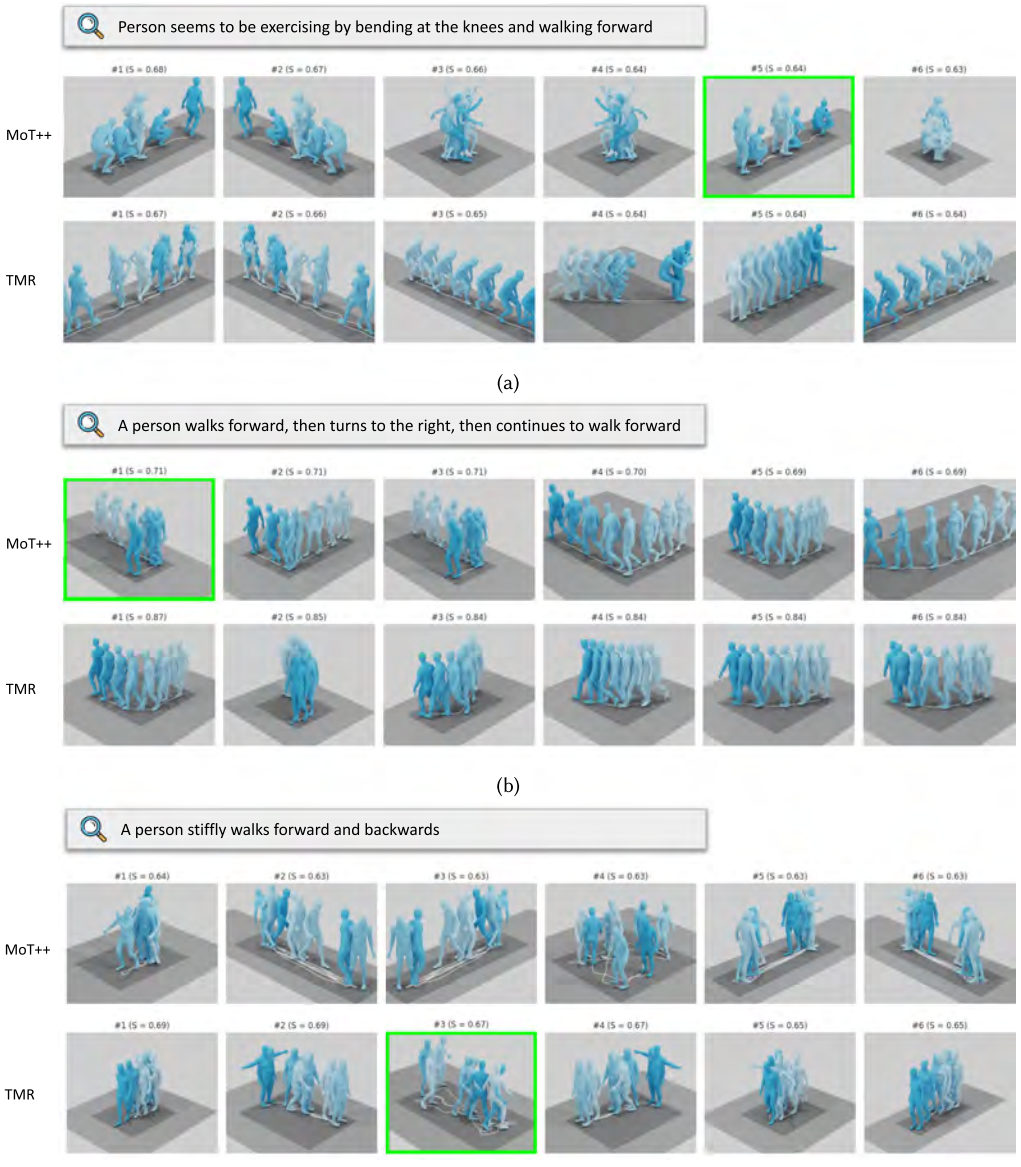



Fig. 5. Qualitative examples on *text-to-motion* retrieval. Samples (a) and (b) show success cases in which MoT++ can find the GT text better describing the motion within the first six results (highlighted in green), while TMR finds it at higher ranks. Sample (c), instead, shows a failure scenario in which MoT++ cannot find the GT motion due to an uncaught discriminative attribute (*stiffly* in this case).

more than 0.95 textual similarity with the others also have the same label. We report both MoT++ with the main hyper-parameter configuration (CCCL 40-100), as well as the interesting case of fully self-supervised learning (MoT++ self), where the teacher model $\mathcal{E}_t^{\text{ref}}$ is not employed. As you may notice, our method reaches the best results on almost all the protocols, achieving the best overall results on the averaged values. Notably, the best result in this scenario is often obtained

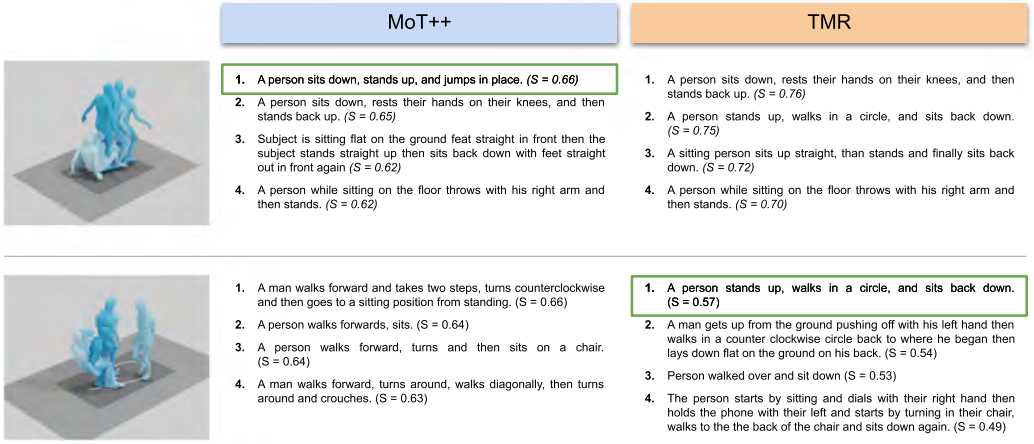



Fig. 6. Qualitative results for *motion-to-text* retrieval. The top example shows how MoT++ can effectively retrieve the GT description in the first position. The second example, instead, shows a challenging case in which TMR is able to catch the correct text, although MoT++ can still retrieve texts that are relevant to at least a subpart of the query motion.

using the "MoT++ self" configuration. This may be attributed to the fact that the text model may introduce some text-specific biases during the optimization of the uni-modal manifolds, which act as slight noise in the motion domain. These results show that our model can effectively learn either a good cross-modal space, where text-to-motion and motion-to-text searches can be effectively performed, and a good uni-modal motion space suitable for the downstream query-by-example application scenarios.

*4.3.6 Visual Inspection of Text-to-Motion Retrieval Results.* In Figure 5, we show some targeted success examples (Figure 5(a) and (b)), as well as some failure case (Figure 5(c)) of our method for text-to-motion retrieval, comparing MoT++ with the state-of-the-art model TMR. Specifically, the first two subfigures show how our model can retrieve motions that better follow the provided fine-grained textual query. In particular, in Figure 5(a), our model better attends to the composite motion requested by the query (*bending the knees* and *walking forward*) in contrast to TMR which, instead, finds people just walking forward as the most relevant results. In Figure 5(b), we can assess the capability of MoT++ to separate actions in the motion that happen in subsequent temporal order (first *walk forward*, then *turn right*, then *walk forward* again). While this sequence is clear in MoT++, this is not always the case in TMR, which retrieves motions where the person simply performs a slight turn to the right. Instead, in Figure 5(c), we show a challenging example where our model, differently from TMR, cannot correctly understand fine-grained textual details, like *stiffly*.

Similarly, in Figure 6, we report success and failure cases for the motion-to-text retrieval scenario. Specifically, the top example shows how our method retrieves the best-matching textual description as the first element, while TMR fails to capture the final action performed in the motion—*jumping*. Instead, in the lower example, we can notice how our model is missing the looping nature of the motion (*the person is sitting, then stands up to walk in a circle, then sits back again*), despite retrieving texts that correctly describe consecutive actions of the query motion (*A person walks forwards, sits*).

## 5 Conclusions

In this work, we incorporated cross-dataset and joint-dataset training for solving text-motion retrieval tasks. We introduced the enhanced motion encoder MoT++ and the CCCL loss, which mitigate the gaps originating from different datasets. Our approach outperforms state-of-the-art

approaches in the joint-dataset or cross-dataset scenarios, evaluated on different combinations of HumanML3D, KITML, and Motion-X datasets. We also demonstrated the generalization abilities and robustness of our approach, which achieves better results on the cross-dataset scenario (e.g., trained on HumanML3D and validated on KITML) in comparison with the narrowly focused single-dataset scenario (e.g., trained and validated on KITML). Future improvements include integrating additional modalities (e.g., video modality), applying pairwise text-to-modality learning to further boost the performance as in LanguageBind [74], or integrating dual-modality regularization similarly to [18].

## References


[1] Emre Aksan, Manuel Kaufmann, Peng Cao, and Otmar Hilliges. 2020. A spatio-temporal transformer for 3D Human motion prediction. arXiv:2004.08692. DOI: https://doi.org/10.48550/ARXIV.2004.08692

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. 2021. ViViT: A video vision transformer. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 6836–6846.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. arxiv:2005.14165. Retrieved from https://arxiv.org/abs/2005.14165

[4] Petra Budikova, Jan Sedmidubsky, and Pavel Zezula. 2021. Efficient indexing of 3D human motions. In *International Conference on Multimedia Retrieval (ICMR)*. ACM, 10–18. Retrieved from https://dl.acm.org/doi/10.1145/3460426.3463646

[5] Qin Cheng, Jun Cheng, Zhen Liu, Ziliang Ren, and Jianming Liu. 2024. A dense-sparse complementary network for human action recognition based on RGB and skeleton modalities. *Expert Systems with Applications* 244 (2024), 1–16. DOI: https://doi.org/10.1016/j.eswa.2023.123061

[6] Yi-Bin Cheng, Xipeng Chen, Junhong Chen, Pengxu Wei, Dongyu Zhang, and Liang Lin. 2021. Hierarchical transformer: Unsupervised representation learning for skeleton-based human action recognition. In *IEEE International Conference on Multimedia and Expo (ICME)*, 1–6.

[7] Yi-Bin Cheng, Xipeng Chen, Dongyu Zhang, and Liang Lin. 2021. Motion-transformer: Self-supervised pre-training for skeleton-based action recognition. In *2nd ACM International Conference on Multimedia in Asia (MMAsia)*. ACM, New York, NY.

[8] Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. 2022. DG-STGCN: Dynamic spatial-temporal modeling for skeleton-based action recognition. arXiv:2210.05895. DOI: https://doi.org/10.48550/ARXIV.2210.05895

[9] Shradha Dubey and Manish Dixit. 2022. A comprehensive survey on human pose estimation approaches. *Multimedia Systems* 29 (2022), 1–29. Retrieved from https://link.springer.com/article/10.1007/s00530-022-00980-0

[10] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. 2021. Clip2video: Mastering video-text retrieval via image clip. arXiv:2106.11097. Retrieved from https://doi.org/10.48550/arXiv.2106.11097

[11] Kent Fujiwara, Mikihiro Tanaka, and Qing Yu. 2024. Chronologically accurate retrieval for temporal grounding of motion-language models. In *18th European Conference on Computer Vision (ECCV)*. Springer-Verlag, Berlin, 323–339. DOI: https://doi.org/10.1007/978-3-031-73636-0_19

[12] Zikai Gao, Peng Qiao, and Yong Dou. 2023. HAAN: Human action aware network for multi-label temporal action detection. In *31st ACM International Conference on Multimedia (MM)*. ACM, New York, NY, 5059–5069. DOI: https://doi.org/10.1145/3581783.3612097

[13] Anindita Ghosh, Noshaba Cheema, Cennet Oguz, Christian Theobalt, and Philipp Slusallek. 2021. Synthesis of compositional animations from textual descriptions. In *IEEE/CVF International Conference on Computer Vision*, 1396–1406.

[14] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. 2022. Generating diverse and natural 3D human motions from text. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5152–5161.

[15] Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. 2022. TM2T: Stochastic and tokenized modeling for the reciprocal generation of 3D human motions and texts. In *European Conference on Computer Vision (ECCV)*. Springer Nature Switzerland, Cham, 580–597.

[16] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. 2020. Action2motion: Conditioned generation of 3d human motions. In *28th ACM International Conference on Multimedia*, 2021–2029.

[17] Tianyu Guo, Hong Liu, Zhan Chen, Mengyuan Liu, Tao Wang, and Runwei Ding. 2022. Contrastive learning from extremely augmented skeleton sequences for self-supervised action recognition. In *36th AAAI Conference on Artificial Intelligence (AAAI)*, 762–770.

[18] Hailang Huang, Zhijie Nie, Ziqiao Wang, and Ziyu Shang. 2024. Cross-modal and uni-modal soft-label alignment for image-text retrieval. In *38th AAAI Conference on Artificial Intelligence*. AAAI Press. DOI: https://doi.org/10.1609/aaai.v38i16.29789

[19] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM TOIS* 20, 4 (2002), 422–446.

[20] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*. PMLR, 4904–4916.

[21] Zi-Fei Jiang, Wei Li, Yan Huang, Yi-Long Yin, C.-C. Jay Kuo, and Jing-Liang Peng. 2023. PESTA: An elastic motion capture data retrieval method. *Journal of Computer Science and Technology* 38 (2023), 867–884. DOI: https://doi.org/10.1007/s11390-023-3140-y

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.

[23] Jihoon Kim, Youngjae Yu, Seungyoun Shin, Taehyun Byun, and Sungjoon Choi. 2022. Learning joint representation of human motion and language. arXiv:2210.15187. Retrieved from https://arxiv.org/abs/2210.15187

[24] Taehoon Kim, ChanHee Kang, JaeHyuk Park, Daun Jeong, ChangHee Yang, Suk-Ju Kang, and Kyeongbo Kong. 2024. Human motion aware text-to-video generation with explicit camera control. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 5081–5090.

[25] Diederik P. Kingma and Max Welling. 2019. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning* 12, 4 (2019), 307–392.

[26] Sumin Lee, Sangmin Woo, Yeonju Park, Muhammad Adi Nugroho, and Changick Kim. 2023. Modality mixer for multi-modal action recognition. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 3298–3307.

[27] Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. 2023. Motion-X: A large-scale 3D expressive whole-body human motion dataset. In *Advances in Neural Information Processing Systems*.

[28] Lilang Lin, Sijie Song, Wenhan Yang, and Jiaying Liu. 2020. MS2L: Multi-task self-supervised learning for skeleton based action recognition. In *28th ACM International Conference on Multimedia (MM)*. ACM, New York, NY, 2490–2498. DOI: https://doi.org/10.1145/3394171.3413548

[29] Yang Liu, Hong Liu, Huaqiu Wang, Fanyang Meng, and Mengyuan Liu. 2024. BCAN: Bidirectional correct attention network for cross-modal retrieval. *IEEE Transactions on Neural Networks and Learning Systems* 35, 10 (2024), 14247–14258. DOI: https://doi.org/10.1109/TNNLS.2023.3276796

[30] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2023. SMPL: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries*, Vol. 2, 851–866.

[31] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2022. CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval and captioning. *Neurocomputing* 508 (2022), 293–304.

[32] Na Lv, Ying Wang, Zhiquan Feng, and Jingliang Peng. 2021. Deep hashing for motion capture data retrieval. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2215–2219. DOI: https://doi.org/10.1109/ICASSP39728.2021.9413505

[33] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. 2019. AMASS: Archive of motion capture as surface shapes. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 5442–5451.

[34] Nicola Messina, Giuseppe Amato, Andrea Esuli, Fabrizio Falchi, Claudio Gennaro, and Stéphane Marchand-Maillet. 2021. Fine-grained visual textual alignment for cross-modal retrieval using transformer encoders. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 17, 4 (2021), 1–23.

[35] Nicola Messina, Davide Alessandro Coccomini, Andrea Esuli, and Fabrizio Falchi. 2022. Transformer-based multi-modal proposal and re-rank for wikipedia image-caption matching. arXiv:2206.10436. Retrieved from https://arxiv.org/abs/2206.10436

[36] Nicola Messina, Jan Sedmidubsky, Fabio Carrara, and Tomáš Rebok. 2023. Text-to-motion retrieval: Towards joint understanding of human motion data and natural language. In *46th International Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, 2420–2425.

[37] Nicola Messina, Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Fabrizio Falchi, Giuseppe Amato, and Rita Cucchiara. 2022. ALADIN: Distilling fine-grained alignment scores for efficient image-text matching and retrieval. In *19th International Conference on Content-based Multimedia Indexing*, 64–70.

[38] Konstantinos Papadopoulos, Enjie Ghorbel, Renato Baptista, Djamila Aouada, and Björn E. Ottersten. 2019. Two-stage RGB-based action detection using augmented 3D poses. In *18th International Conference on Computer Analysis of Images and Patterns (CAIP)*, Vol. 11678, Springer, 26–35.

[39] Mathis Petrovich, Michael J. Black, and Gül Varol. 2021. Action-conditioned 3D human motion synthesis with transformer VAE. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 10985–10995.

[40] Mathis Petrovich, Michael J. Black, and Gül Varol. 2022. TEMOS: Generating diverse human motions from textual descriptions. In *European Conference on Computer Vision (ECCV)*. Springer Nature Switzerland, Cham, 480–497.
[41] Mathis Petrovich, Michael J. Black, and Gül Varol. 2023. TMR: Text-to-motion retrieval using contrastive 3D human motion synthesis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 9488–9497.
[42] Matthias Plappert, Christian Mandery, and Tamim Asfour. 2016. The KIT motion-language dataset. *Big Data* 4, 4 (2016), 236–252.
[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. arXiv:2103.00020. Retrieved from https://doi.org/10.48550/arXiv.2103.00020
[44] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1, 8 (2019), 9.
[45] Jan Sedmidubsky, Fabio Carrara, and Giuseppe Amato. 2023. SegmentCodeList: Unsupervised representation learning for human skeleton data retrieval. In *45th European Conference on Information Retrieval (ECIR)*. Springer, Cham, 110–124. DOI: https://doi.org/10.1007/978-3-031-28238-6_8
[46] Jan Sedmidubsky, Petr Elias, Petra Budikova, and Pavel Zezula. 2021. Content-based management of human motion data: Survey and challenges. *IEEE Access* 9 (2021), 64241–64255. DOI: https://doi.org/10.1109/ACCESS.2021.3075766
[47] Muhammad Bilal Shaikh, Douglas Chai, Syed Muhammad Shamsul Islam, and Naveed Akhtar. 2024. From CNNs to transformers in multimodal human action recognition: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications* 20 (2024), 1–24. Retrieved from https://dl.acm.org/doi/10.1145/3664815
[48] Nina Shvetsova, Brian Chen, Andrew Rouditchenko, Samuel Thomas, Brian Kingsbury, Rogerio S. Feris, David Harwath, James Glass, and Hilde Kuehne. 2022. Everything at once-multi-modal fusion transformer for video retrieval. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20020–20029.
[49] Nyle Siddiqui, Praveen Tirupattur, and Mubarak Shah. 2023. DVANet: Disentangling view and action features for multi-view action recognition. arXiv:2312.05719. Retrieved from https://arxiv.org/abs/2312.05719
[50] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, Vol. 33, 16857–16867.
[51] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. 2018. Spatio-temporal attention-based LSTM networks for 3D action recognition and detection. *IEEE Transactions on Image Processing* 27, 7 (2018), 3459–3471. DOI: https://doi.org/10.1109/TIP.2018.2818328
[52] Wenfeng Song, Tangli Chu, Shuai Li, Nannan Li, Aimin Hao, and Hong Qin. 2023. Joints-centered spatial-temporal features fused skeleton convolution network for action recognition. *IEEE Transactions on Multimedia* (2023), 1–15. DOI: https://doi.org/10.1109/TMM.2023.3324835
[53] Shengkai Sun, Daizong Liu, Jianfeng Dong, Xiaoye Qu, Junyu Gao, Xun Yang, Xun Wang, and Meng Wang. 2023. Unified multi-modal unsupervised representation learning for skeleton-based action understanding. In *31st ACM International Conference on Multimedia (MM)*. ACM, 2973–2984.
[54] Yansong Tang, Xingyu Liu, Xumin Yu, Danyang Zhang, Jiwen Lu, and Jie Zhou. 2022. Learning from temporal spatial cubism for cross-dataset skeleton-based action recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications* 18, 2 (2022), 1–24.
[55] Ömer Terlemez, Stefan Ulbrich, Christian Mandery, Martin Do, Nikolaus Vahrenkamp, and Tamim Asfour. 2014. Master Motor Map (MMM)—Framework and toolkit for capturing, representing, and reproducing human motion on humanoid robots. In *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 894–901.
[56] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-Or, and Amit Haim Bermano. 2023. Human motion diffusion model. In *11th International Conference on Learning Representations (ICLR)*. Retrieved from https://openreview.net/forum?id=SJ1kSyO2jwu
[57] Xiaoyan Tian, Ye Jin, Zhao Zhang, Peng Liu, and Xianglong Tang. 2023. Spatial-temporal graph transformer network for skeleton-based temporal action segmentation. *Multimedia Tools and Applications* (2023), 1–25. DOI: https://doi.org/10.1007/s11042-023-17276-8
[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, Vol. 30.
[59] Guoquan Wang, Hong Liu, Tianyu Guo, Jingwen Guo, Ti Wang, and Yidi Li. 2023. Self-supervised 3D skeleton representation learning with active sampling and adaptive relabeling for action recognition. In *IEEE International Conference on Image Processing (ICIP)*, 56–60.
[60] Qiang Wang, Junlong Du, Ke Yan, and Shouhong Ding. 2023. Seeing in flowing: Adapting CLIP for action recognition with motion prompts learning. In *31st ACM International Conference on Multimedia (MM)*. ACM, 5339–5347. DOI: https://doi.org/10.1145/3581783.3612490
[61] Libo Weng, Weidong Lou, and Fei Gao. 2024. Language guided graph transformer for skeleton action recognition. In *Neural Information Processing*. Springer Nature Singapore, Singapore, 283–299.

[62] Sheng Yan, Mengyuan Liu, Yong Wang, Yang Liu, and Hong Liu. 2024. MLP: Motion label prior for temporal sentence localization in untrimmed 3D human motions. *IEEE Transactions on Circuits and Systems for Video Technology* 34 (2024), 11535–11550. Retrieved from https://ieeexplore.ieee.org/document/10584551

[63] Sheng Yan, Yang Liu, Haoqiang Wang, Xin Du, Mengyuan Liu, and Hong Liu. 2023. Cross-modal retrieval for motion and text via DropTriple loss. In *5th ACM International Conference on Multimedia in Asia (MMAsia)*. ACM, New York, NY, 1–7.

[64] Sheng Yan, Yong Wang, Xin Du, Hongchang Jin, and Mengyuan Liu. 2024. Improving fine-grained understanding for retrieval in human motion and text. *IEEE Signal Processing Letters* (2024), 1–5. DOI: https://doi.org/10.1109/LSP.2024.3425283

[65] YangYang, Guangjun Liu, and Xuehao Gao. 2022. Motion guided attention learning for self-supervised 3D human action recognition. *IEEE Transactions on Circuits and Systems for Video Technology* (2022), 1–13. DOI: https://doi.org/10.1109/TCSVT.2022.3194350

[66] YangYang, Haoyu Shi, and Huaiwen Zhang. 2024. Hierarchical semantics alignment for 3D human motion retrieval. In *47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, New York, NY, 1083–1092.

[67] Qing Yu, Mikihiro Tanaka, and Kent Fujiwara. 2024. Exploring vision transformers for 3D human motion-language models with motion patches. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 937–946. DOI: https://doi.org/10.1109/CVPR52733.2024.00095

[68] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. 2023. T2M-GPT: Generating human motion from textual descriptions with discrete representations. arXiv:2301.06052. DOI: https://doi.org/10.48550/ARXIV.2301.06052

[69] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. 2022. MotionDiffuse: Text-driven human motion generation with diffusion model. arXiv:2208.15001. DOI: https://doi.org/10.48550/ARXIV.2208.15001

[70] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Revisiting visual representations in vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5579–5588.

[71] Yaqi Zhang, Di Huang, Bin Liu, Shixiang Tang, Yan Lu, Lu Chen, Lei Bai, Qi Chu, Nenghai Yu, and Wanli Ouyang. 2023. MotionGPT: Finetuned LLMs are general-purpose motion generators. arXiv:2306.10900. Retrieved from https://doi.org/10.48550/arXiv.2306.10900

[72] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D. Manning, and Curtis P. Langlotz. 2020. Contrastive learning of medical visual representations from paired images and text. arXiv:2010.00747. Retrieved from https://arxiv.org/abs/2010.00747

[73] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. 2019. On the continuity of rotation representations in neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5745–5753.

[74] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, HongFa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, et al. 2024. LanguageBind: Extending video-language pretraining to N-modality by language-based semantic alignment. In *International Conference on Learning Representations (ICLR)*.

[75] Mu Zhu. 2004. Recall, precision and average precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo* 2, 30 (2004), 6.

[76] Tianming Zhuang, Zhen Qin, Yi Ding, Fuhu Deng, LeDuo Chen, Zhiguang Qin, and Kim-Kwang Raymond Choo. 2024. Temporal refinement graph convolutional network for skeleton-based action recognition. *IEEE Transactions on Artificial Intelligence* 5, 4 (2024), 1–14. Retrieved from https://ieeexplore.ieee.org/document/10310028

Received 27 June 2024; revised 3 June 2025; accepted 5 June 2025