HaLP: Hallucinating Latent Positives for Skeleton-based Self-Supervised Learning of Actions

Anshul Shah[1] Aniket Roy[1*] Ketul Shah[1*] Shlok Mishra[2]
David Jacobs[2,3] Anoop Cherian[4] Rama Chellappa[1]
[1]Johns Hopkins University [2]University of Maryland, College Park [3]Meta [4]MERL
{ashah95, aroy28, kshah33, rchella4}@jhu.edu {shlokm, dwj}@umd.edu cherian@merl.com

Abstract

Supervised learning of skeleton sequence encoders for action recognition has received significant attention in recent times. However, learning such encoders without labels continues to be a challenging problem. While prior works have shown promising results by applying contrastive learning to pose sequences, the quality of the learned representations is often observed to be closely tied to data augmentations that are used to craft the positives. However, augmenting pose sequences is a difficult task as the geometric constraints among the skeleton joints need to be enforced to make the augmentations realistic for that action. In this work, we propose a new contrastive learning approach to train models for skeleton-based action recognition without labels. Our key contribution is a simple module, HaLP *– to Hallucinate Latent Positives for contrastive learning. Specifically, HaLP explores the latent space of poses in suitable directions to generate new positives. To this end, we present a novel optimization formulation to solve for the synthetic positives with an explicit control on their hardness. We propose approximations to the objective, making them solvable in closed form with minimal overhead. We show via experiments that using these generated positives within a standard contrastive learning framework leads to consistent improvements across benchmarks such as NTU-60, NTU-120, and PKU-II on tasks like linear evaluation, transfer learning, and kNN evaluation. Our code can be found at https://github.com/anshulbshah/HaLP.*

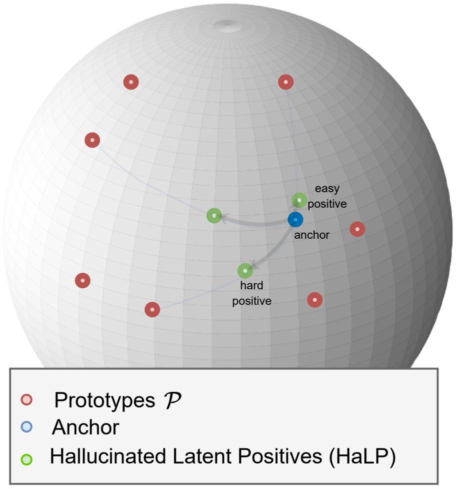
easy positive
anchor
hard positive
○ Prototypes $\mathcal{P}$
○ Anchor
○ Hallucinated Latent Positives (HaLP)

Figure 1. HaLP: We propose an approach to hallucinate latent positives for use within a contrastive learning pipeline. Our approach works as follows: 1) We extract prototypes which succinctly represent the data at a particular step in training, 2) We randomly select a prototype from the prototype set, 3) Our approach then determines an optimal vector which when added to the anchor can generate positives of varying hardness. We generate a number of positives using this approach which are then used within a contrastive learning pipeline to train a model without labels.

1. Introduction

Recognizing human actions from videos is of immense practical importance with applications in behavior understanding [52], medical assistive applications [5], AR/VR applications [26] and surveillance [12]. Action recognition has been an active area of research [29] with a focus on temporal understanding [14], faster and efficient models for understanding complex actions [38], etc. Most works in the past have focused on action recognition from appearance information. But recent methods have shown the advantages of using pose/skeleton information as a separate cue with benefits in robustness to scene and object biases [34, 35, 53], reduced privacy concerns [27], apart from succinctly representing human motion [24]. However, annotating videos for skeleton-based action recognition is an arduous task and

[*] Aniket Roy and Ketul Shah contributed equally

is difficult to scale. Prior work in self-supervised learning has shown advantages of learning without labels - including improved transfer performance [6, 22], robustness to domain shift [15, 51] and noisy labels [11, 16]. Inspired by these methodologies, we propose a new approach for skeleton-based action recognition without using labels.

There have been several interesting approaches to tackling self-supervision for skeleton sequences. Methods like [41, 59] have proposed improved pretext tasks to train models. Image-based self-supervised learning has shown impressive success using contrastive learning (CL)-based losses. Inspired by these, some recent approaches [37, 48] have successfully applied CL to skeleton sequences, with modifications such as data augmentations [48], use of multi-modal cues [37], etc. The success of CL for a problem is closely tied to the data augmentations used to create the positives and the quality and number of negatives used to offer contrast to the positives [6, 22]. While various works have tried focusing on negatives for improving the performance of Skeleton-SSL models, augmenting skeleton sequences is more difficult. Unlike images, skeletons are geometric structures, and devising novel data augmentations to craft new positives is an interesting but difficult task.

In this work, we address the question of whether we can hallucinate new positives in the latent space (Fig. 1) for use in a CL framework. Our approach, which we call HaLP: **Ha**llucinating **L**atent **P**ositives has dual benefits; generating positives in latent space can reduce reliance on hand-crafted data augmentations. Further, it allows for faster training than using multiple-view approaches [3, 4], which incur significant overheads. Recall that, CL trains a model by pulling two augmented versions of a skeleton sequence close in the latent space while pushing them far apart from the negatives. Our key idea in this work is to hallucinate new positives, thus exploring new parts of the latent space beyond the query and key to improve the learning process. We introduce two new components to the CL pipeline. The first extracts prototypes from the data which succinctly represent the high dimensional latent space using a few key *centroids* by clustering on the hypersphere. Next, we introduce a Positive Hallucination (PosHal) module. Naïvely exploring the latent space might lead to sub-optimal positives or even negatives. We overcome this by proposing an objective function to define hard positives. The intuition is that the similarity of the generated positives and real positives should be minimized such that both have identical closest prototypes. Since solving this optimization problem for each step of training could be expensive, we propose relaxations that let us derive a closed-form expression to find the hardest positive along a particular direction defined by a randomly selected prototype. The final solution involves a spherical linear interpolation between the anchor and a randomly selected prototype with explicit control of hardness of the generated positives.

We experimentally verify the efficacy of HaLP approach by experiments on standard benchmark datasets: NTU RGB-D 60, NTU RGB-D 120, and PKU-MMD II and notice consistent improvements over state-of-the-art. For example, on the linear evaluation protocol, we obtain +2.3%, +2.3%, and +4.5% for cross-subject splits of the NTU-60, NTU-120, and PKU-II datasets respectively. Using our module with single-modality training leads to consistent improvements as well. Our model trained on single modality, obtains results competitive to a recent approach [37] which uses multiple modalities during training while being 2.5x faster.

In summary, the following are our main contributions:

1. We propose a new approach, HaLP which hallucinates latent positives for use in a skeleton-based CL framework. To the best of our knowledge, we are the first to analyze the generation of positives for CL in latent-space.
2. We define an objective function that optimizes for generating hard positives. To enable fast training, we propose relaxations to the objective which lets us derive closed-form solutions. Our approach allows for easy control of the hardness of the generated positives.
3. We obtain consistent improvements over the state-of-the-art methods on all benchmark datasets and tasks. Our approach is easy to use and works in uni-modal and multi-modal training, bringing benefits in both settings.

2. Related Work

Self-Supervised Learning: Much of the progress in representation learning has been in the supervised paradigm. But owing to huge annotations costs and an abundance of unlabeled data, pretraining models using self-supervised techniques have received a lot of attention lately. Early works in computer vision developed pretext tasks such as predicting rotation [17], solving jigsaw puzzle [41], image colorization [59], and temporal order prediction [39] to learn good features. Contrastive learning [19, 20] is one such pretext task that relies on instance discrimination - the goal is to classify a set of positives (augmented version of the same instance) against a set of unrelated negatives which helps the model learn good features. Recent works have demonstrated exceptional performance using these techniques in a variety of domains [23], including images, videos, graphs, text, etc. SimCLR [6] and MoCo [22] frameworks have been very popular due to their ease of use and general applicability. Recently, several non-contrastive learning approaches like SwAV [3], DINO [4], MAE [21] have shown promising performance but CL still offers complementary benefits [30, 32]. In this work, we focus on CL objectives which have been shown to be superior to non-contrastive ones for the task of skeleton-based SSL [37].

Self-Supervised Skeleton/Pose-based Action Recognition: Supervised Skeleton-based action recognition has received a lot of attention due to its wide applicability. Research in this field has led to new models [13, 54] and learning representations [10]. Several works have been proposed to explore the benefits of self-supervision in this space. Some prior works have used novel pretext tasks to learn representations including skeleton colorization [55], displacement prediction [28], skeleton inpainting [61], clustering [47] and Barlow twins-based learning [58]. Another area of interest has been on how to best represent the skeletons for self-supervised learning. Some works have explored better encoders and training schemes for skeletal sequences like Hierarchical transformers for better spatial hierarchical modeling [9], local-global mechanism and better handling of multiple persons in the scene [28] or use of multiple pretext tasks at different levels [8]. Complementary to innovations in the modeling of skeletons, various works have used ideas from contrastive learning for self-supervised learning of skeleton representations [33, 37, 42, 48] and have shown exceptional performance. Augmentations are crucial to contrastive learning and various works [18, 48] have studied techniques to augment skeletal data. Others have explored the use of additional information during training like multiple skeleton representations [48], multiple skeleton modalities [33, 37], local-global correspondence and attention [28]. We work with CL, owing to simplicity and strong representations. We use the same training protocols and encoders as CMD [37], but show how our proposed approach can hallucinate latent positives which can be generated using a very fast plug-and-play module. Our approach shows significant improvements on both single-modality training and multi-modal training.

Role of positives and negatives for CL Prior work has shown that the use of a large number of negatives [22] and hard negatives play an important role in contrastive learning while false negatives impact learning. DCL [11] helps account and correct for false negatives, while HCL [43] also allows for control of hard negatives. MMCL [44] uses a max-margin framework to handle hard and false negatives while [15] uses texture-based negative samples to create hard negatives. While there has been some focus on generating better positives for SSL, most approaches only consider generating better input views or to generate hard negatives. Mixup [57], Manifold mixup [50], and their variants like CutMix [56] have been very popular in supervised learning setups to regularize and augment data. Various approaches have proposed novel approaches to use these ideas in a self-supervised setting. M-Mix [60] uses adaptive negative mixup in the input image space, multi-modal mixup [46] generates hard negatives by using multimodal information, [45] makes the model aware of the soft similarity between generated images to learn robust representations. In contrast, i-mix [31] creates a virtual label space by training a non-parametric classifier. Closely related to our approach is MoCHI [25] which creates hard negatives for a MoCo-based framework by mixup of latent space features of positives and negatives. Unlike these works, our focus in this work is to create *positives*. Instead of relying on heuristics, we instead pose generation of hard positives as an optimization problem and propose relaxation to solve this problem efficiently.

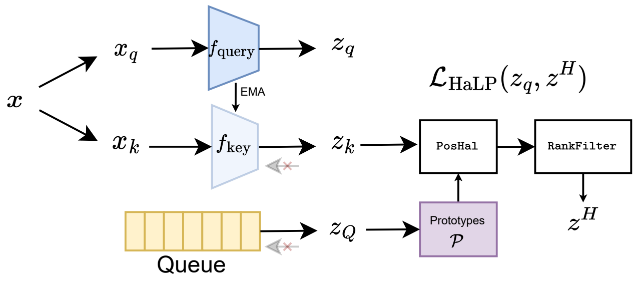

Figure 2. Overall approach to hallucinate positives. We work with a MoCo-based framework. Query and Key encoders represent the input skeleton sequence in the latent space. The queue is maintained by using past key features. The `PosHal` module hallucinates positives using the anchor and prototypes extracted from the queue. Positives satisfying the rank filter constraint are retained and used to calculate the HaLP loss. The model is trained with a weighted combination of the standard CL loss and HaLP loss.

3. Method

In this work, we hallucinate new positives and use them within a self-supervised learning pipeline to learn better representations for skeleton-based action recognition. Fig. 2 presents an overview of our training pipeline.

3.1. Preliminaries

Problem setup. We are given a dataset $\mathcal{D}$ of unlabeled 3D skeleton sequences. We wish to learn without labels, an encoder E to represent a sequence from this dataset in a high dimensional space such that the encoder can then later be adapted to a task where little training data is available. Specifically, let the skeleton sequence be $x \in \mathbb{R}^{3\times F\times M\times P}$, where F denotes the number of frames in the sequence, M is the number of joints, P denotes the number of people in the scene, and the first dimension encodes the coordinates $[\mathrm{x}, \mathrm{y}, \mathrm{z}]$. The encoder E takes this sequence and generates a representation that is used for the downstream task. Prior work [37, 48] in skeleton-based SSL has worked with various modalities like bones, joints, and motion, which are extracted from raw joint coordinates. In contrast, our approach can be applied to both single-modality and multi-modality training.

Contrastive learning preliminaries. The key idea in CL is to maximize the similarity of two views (*positives*) of a skeleton sequence of a video and push it away from features of other sequences (*negatives*). The similarities are computed in a high-dimensional representation space. We adopt the MoCo framework [7, 22], which has been successfully applied to various image, video, and skeleton self-supervised learning tasks. The input skeleton sequence x is first transformed by two different random cascade of augmentations $T_q, T_k \in \mathcal{T}$ to generate positives: query x_q and key x_k. MoCo uses separate encoders, query encoder E_q and key encoder E_k to generate L2-normalized representations z_q, $z_k \in \mathbb{R}^D$ on the unit hypersphere. While the E_q is trained using backpropagation, E_k is updated through a moving average of weights of E_q. In addition to encoding the positives, MoCo maintains a queue $Q = \{z_i^Q\}$ with $z_i^Q \in \mathbb{R}^D$. The queue contains L encoded keys from the most recent iterations of training. The elements of the queue are used as negatives in the contrastive learning process. To train the model, an InfoNCE objective function is used to update the parameters of E_q. The loss function used is given below:

$$\mathcal{L}_{\text{CL}} = -\log \frac{\exp(z_q^\top z_k/\tau)}{\exp(z_q^\top z_k/\tau) + \sum_{i=1}^{L} \exp\left(z_q^\top z_i^Q/\tau\right)}, \quad (1)$$

where τ is the temperature hyperparameter. As discussed in Sec. 2, there has been a lot of work in generating, and finding hard negatives which help in learning better representations. Choosing the right augmentations is critical to the learning process and works in the past have shown that including multiple views [3, 4] helps in the learning process. But, these works craft new positives in the input space which can significantly increase the training time due to the additional forward/backward passes to the encoder.

3.2. Hallucinating Positives

We now present our lightweight module, which can hallucinate new positives in the feature space. This has two-fold advantages: 1) This can reduce the burden on designing new data augmentations, 2) Since we hallucinate positives directly in the feature space, we do not need to backpropagate their gradients to the encoder, thus saving on expensive forward/backward passes during training. This is especially amenable to the MoCo framework [22] where z_k does not have any gradients associated with it. Our newly generated positives z_i^H play the same role as z_k, except that these are hallucinated synthetic positives instead of being obtained through a real skeleton sequence.

Analogous to hard negatives [25, 43, 44], we define hard positives as samples which lie far from an anchor positive in latent space but have the same semantics. Thus, we desire that our hallucinated positives should be diverse with varying amount of hardness to provide a good training signal. Further, they should have a high semantic overlap with the real positives. These are two conflicting requirements. Therefore, it is important to achieve a balance between the difficulty and similarity to the original positives when generating positive data points. This will ensure that the generated points remain valid true positives and do not introduce false positives that may hinder the training process.

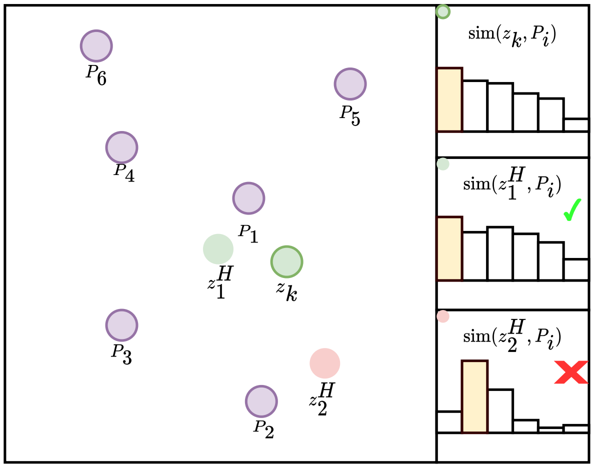

Figure 3. Intuition behind Eq. (2). Our objective function enforces the constraint that the hallucinated positives and the original anchor (z_k) have the same closest prototype (P_1) while minimizing the similarity to z_k. For example, z_2^H does not satisfy the constraint while z_1^H does.

Our key intuition behind generating synthetic positives is that given the current encoded (anchor) key z_k, we can explore the high dimensional space around it to find locations that can plausibly be reached by the encoder for closely related skeleton sequences.

We define $\mathcal{P} = \{P_1, \cdots, P_N\}$ as N cluster centroids of the data. Based on our desiderata, we can formulate the following objective to find hard positives :

$$\begin{aligned} z^* &= \arg \min_{z \in \mathbb{S}^{D-1}} \ \text{sim}(z, P^*_{z_k}) \\ \text{s.t.} & \quad \text{sim}(z, P^*_{z_k}) \geq \text{sim}(z, P), \forall P \in \mathcal{P} \backslash \left\{P^*_{z_k}\right\}, \end{aligned} \quad (2)$$

where sim() is the cosine similarity, $\mathbb{S}^{D-1}$ represents the unit hypershpere, and $P^*_{z_k} = \arg\max_{P \in \mathcal{P}} \text{sim}(z_k, P)$ is the prototype closest to z_k (our anchor).

Intuitively, we want to generate hard positives which are far from the anchor but have the same closest prototype as the anchor. We call the constraint our Rank Filter which is visualized in Fig. 3.

Note that one could use Riemannian optimization solvers (e.g., PyManOpt [49]) to solve this problem. However, these are iterative and we desire quick solutions as such optimization needs to be done on every data sample. We propose simplifications to the above objective in the following subsections towards deriving a computationally cheap closed-form solution.

3.3. Restricting the search space

In this section, we make some simplifications to the objective in Eq. (2). First, we define a new positive with the following equation:

$$z = \text{proj}(z_k + d), \tag{3}$$

where z_k is an anchor view to generate the positive, $d \in \mathbb{R}^D$ is the step taken, and $\text{proj}(z') = z'/\|z'\|$. The normalization step ensures that the generated point lies on the unit hypersphere (like z_k and z_q). To constrain the search space for the hallucinated positive, we propose to restrict d towards one of the prototypes P_{sel} selected at random. This helps us relax the search space while moving towards parts of the space which are occupied by instances from the dataset which can help generate hard positives. Thus, we restrict the search space instead of searching for a z (as in Eq. (2)). We borrow ideas from Manifold Mixup [50] to define intermediate points along the geodesic joining z_k and P_{sel}. Since we are working with points on the hypersphere, we have the following search space:

$$\begin{gathered} d(t, P_{\text{sel}}, z_k) = \frac{\sin(1-t)\Omega}{\sin\Omega} z_k + \frac{\sin(t\Omega)}{\sin\Omega} P_{\text{sel}} - z_k, \\ \text{where } t \in [0,1], \cos\Omega = {P_{\text{sel}}}^\top z_k \text{ and } \Omega \in [0, \pi] \end{gathered} \tag{4}$$

We now obtain hard positive using $z^* = z_k + d(t^*, P_{\text{sel}}, z_k)$. The optimal value t^* is calculated by the following modified objective function on the restricted search space.

$$\begin{aligned} & t^* = \arg\min_{t\in[0,1]} \text{sim}(z, P^*_{z_k}), \text{where} \\ & z = z_k + d(t, P_{\text{sel}}, z_k) \\ & \text{s.t.} \quad \text{sim}(z, P^*_{z_k}) \geq \text{sim}(z, P_j), P_j \in \mathcal{P} \end{aligned} \tag{5}$$

While Eq. (5) restricts the search space, this still involves solving an optimization problem for each point. Next, we make another simplifying approximation.

3.4. Optimal solutions using pair of prototypes

Instead of solving for the ranking objective during optimization, we solve for the following

$$\begin{aligned} & t^* = \arg\min_{t\in[0,1]} \text{sim}(z, P^*_{z_k}), \text{where} \\ & z = z_k + d(t, P_{\text{sel}}, z_k) \\ & \text{s.t.} \quad \text{sim}(z, P^*_{z_k}) \geq \text{sim}(z, P_{\text{sel}}), \end{aligned} \tag{6}$$

This modified objective effectively solves Eq. (5) assuming just two prototypes $\{P_{\text{sel}}, P^*_{z_k}\}$. This modification lets us derive a closed-form solution to this equation.

Algorithm 1 HaLP : Halucinating Latent Positives

```
**Input:** $x_k$, $x_q$, $E_k$, $E_q$, queue $Q$
**Output:** $\mathcal{L}_{\text{HaLP}}$
# Extract key and query features
$z_k, z_q = E_k(x_k), E_q(x_q)$
# Cluster most recent queue elements into N prototypes
$\mathcal{P} = \texttt{sphere\_cluster}(\texttt{topK}(Q))$
# Find the closest prototype to key
$P^*_{z_k} = \arg\max_P \{\text{sim}(z_k, P_j)\}_j$
# Select a prototype to step towards
$P_{\text{sel}} = \texttt{random}(\mathcal{P})$
# Determine how far we can move from key while still being a member of $P^*_{z_k}$
$t^*$ using Eq. (7)
# Control hardness using $\lambda$
$t_c \sim \texttt{uniform}(0, \lambda t^*)$
# Generate hallucinated positive of $z_k$ (Eq. (4))
${z_i'}^H = z_k + d(t_c, P_{\text{sel}}, z_k)$
# Apply rank-filter to discard generated positives not satisfying constraint in Eq. (2)
$z_i^H = \texttt{rank\_filter}({z_i'}^H)$
# Compute $\mathcal{L}_{\text{HaLP}}$ using Eq. (8)
return $\mathcal{L}_{\text{HaLP}}$
```

$$\begin{aligned} t^* &= \frac{1}{\Omega} \arctan\left(\frac{\sin\Omega}{\kappa + \cos\Omega}\right), \text{where} \\ \kappa &= \frac{1 - {P_{\text{sel}}}^\top P^*_{z_k}}{{z_k}^\top (P^*_{z_k} - P_{\text{sel}})}, \end{aligned} \tag{7}$$

Intuitively, t^* restricts the part of the geodesic between z_k and P_{sel} which are good positives. We then generate the new positives as $z_i^H = z_k + d(t_c, P_{\text{sel}}, z_k)$ where $t_c \sim \texttt{uniform}(0, \lambda t^*)$ where the fixed scalar λ allows us to control the level of hardness of the generated positives. `PosHal` (Fig. 2) uses this strategy to efficiently generate positives for a given batch. Since our relaxation Eq. (6) considers only two prototypes, the generated positives might not satisfy the original ranking constraint (Eq. (2)). Thus, we pass the generated positives through the rank-filter to obtain the final set of filtered positives (Fig. 2).

3.5. How to obtain prototypes

The prototypes $\mathcal{P}$ could intuitively represent various classes, action attributes, etc. Since we are working with points on the unit hypersphere, $\mathbb{S}^{D-1}$, we propose to use k-Means clustering on the hypersphere manifold with the associated Riemannian metric. Since the queue gets updated in a first-in-first-out fashion, we use the `topK` most recent elements of the queue to obtain the prototypes. Next, we calculate the similarities of the z_k to each of the prototypes.

Table 1. Results on linear evaluation: Our proposed module HaLP achieves state-of-the-art results and improves the performance on NTU-60 x-sub dataset by 2.3%, NTU-120 x-sub by 2.3% and PKU-II by 4.5%. We also show significant improvements over the single modality Baseline. Our proposed model is a lightweight module, with minimal overheads and helps in achieving strong performances across various datasets.

Method	NTU-60		NTU-120		PKU-II
	x-sub	x-view	x-sub	x-set	x-sub
Additional training modalities or encoders					
ISC [48]	76.3	85.2	67.1	67.9	36.0
CrosSCLR-B [37]	77.3	85.1	67.1	68.6	41.9
CMD [37]	79.8	86.9	70.3	71.5	43.0
HaLP + CMD	**82.1**	**88.6**	**72.6**	**73.1**	**47.5**
Training using only joint					
LongT GAN [61]	39.1	48.1	-	-	26.0
MS^{2}L [36]	52.6	-	-	-	27.6
P&C [47]	50.7	76.3	42.7	41.7	25.5
AS-CAL [42]	58.5	64.8	48.6	49.2	-
H-Transformer [9]	69.3	72.8	-	-	-
SKT [58]	72.6	77.1	62.6	64.3	-
GL-Transformer [28]	76.3	83.8	66.0	68.7	-
SeBiReNet [40]	-	79.7	-	-	-
AimCLR [18]	74.3	79.7	-	-	-
Baseline	78.0	85.5	69.1	69.8	42.9
HaLP	**79.7**	**86.8**	**71.1**	**72.2**	**43.5**

3.6. Loss function

The final step in the proposed approach involves using these generated points to train the model. Note that these generated points do not have any gradient through them since in Eq. (3), both z_k and d are detached from the computational graph. Thus, we use the generated points as *hallucinated keys* in the MoCo framework. We train our models using the following weighted loss

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CL}} + \mu \mathcal{L}_{\text{HaLP}} \quad \text{where}$$
$$\mathcal{L}_{\text{HaLP}} = -\frac{1}{G_{\text{filtered}}} \sum_{i}^{G_{\text{filtered}}} z_q^\top z_i^H / \tau \tag{8}$$

where G_{filtered} is the number of filtered positives. Algorithm 1 summarizes our entire approach.

4. Experiments

4.1. Datasets:

NTU RGB+D 60: NTU-60 is a large-scale action recognition dataset consisting of 56880 action sequences belonging to 60 categories performed by 40 subjects. The dataset consists of multiple captured modalities, including appearance, depth and Kinect-v2 captured skeletons. Following prior work in skeleton-based action recognition, we only work with the skeleton sequences. These sequences consist of 3D coordinates of 25 skeleton joints. The dataset is used with two protocols: Cross-subject: where half of the subjects are present in the training and the rest for the test; Cross-View where two of the camera's views are used for training and the rest for the test.

NTU RGB+D 120: This dataset was collected as an extension of the NTU RGB+D 60 dataset. This dataset extends the number of actions from 60 to 120, and the number of subjects from 40 to 10 and includes a total of 32 camera setups with varying distances and backgrounds. The dataset contains 114480 sequences. In addition to a Cross-Subject protocol, The dataset proposes a cross-setup protocol as a replacement for cross-view to account for changes in camera distance, views, and backgrounds.

PKU MMD - II: is a benchmark multi-modal 3D human action understanding dataset. Following prior works, we work with the phase 2 dataset which is made quite challenging due to large view variations. Like NTU-60/120, we work with the Kinect-v2 captured sequences provided with the dataset. We use the cross-subject protocol, which has 5332 sequences for the train and 1613 for the test set.

4.2. Implementation details

Encoder models E: For a fair comparison with recent state-of-the-art [37], we make use of a Bidirectional GRU-based encoder. Precisely, it consists of 3D bidirectional recurrent layers with a hidden dimension of 1024. The encoder is followed by a projection layer, and L2 normalized features from the projection layer are used for self-supervised pre-training. As is common in self-supervised learning, pre-MLP features are used for the downstream task. Authors in [37] have shown the benefit of a BiGRU encoder over graph convolutional networks or transformer-based models. We use the same preprocessing steps as prior work where a maximum of two actors are encoded in the sequence, and inputs corresponding to the non-existing actor are set to zero. Inputs to the model are temporally resized to 64 frames.

Baseline: For a fair comparison with prior works using a single modality during training, we also compare our approach to 'baseline'. This essentially implements a single modality, vanilla MoCo using CMD [37] framework without the cross-modal distillation loss [37] or the hallucinated positives. This baseline demonstrates the benefit of CL for this problem using the joint modality alone.

Multimodal training: Our plug-and-play approach is not dependent on the modalities used during training. Unless otherwise specified, we use a single modality (Joint) during training. For multi-modality training experiments, we use the CMD [37] framework and hallucinate per-modality positives. We call this approach HaLP + CMD. Note that using cross-modal positives could lead to further improvements, but we leave this extension to future work to keep our approach general.

We follow the same pre-training hyperparameters as ISC [37, 48]. The models are trained with SGD with a momentum of 0.9 and weight decay of 0.0001. We use a batch size of 64 and a learning rate of 0.01. Models are pretrained for 450 and 1000 epochs on NTU-60/120 and PKU, respectively. The queue size used is 16384. We use the same skeleton augmentations as ISC and CMD. These include pose jittering, shear augmentation and temporal random resize crop.

HaLP specific implementation details: We generate 100 positives per anchor. We use `Geomstats` [1] for k-Means clustering on the hypersphere with a tolerance of 1E-3 and initial step size of 1.0. The prototypes are updated every five iterations. 256 (K) most recent values of the queue were used to cluster. $\lambda = 0.8$ is used for all experiments. We use $\mu = 0$ for the first 200 epochs and $\mu = 1$ for the rest. Wandb [2] was used for experiment tracking.

Evaluation protocols: We evaluate the models on three standard downstream tasks: Linear evaluation, kNN evaluation and transfer learning. In Sec. 4.3 we describe the protocols followed by our results. Additional implementation details and experiments are present in the supplementary material.

4.3. Comparison with state-of-the-art

Linear Evaluation: Here, we freeze the self-supervised pre-trained encoder and attach an MLP classifier head to it. The model is then trained with labels on the dataset. We pretrain our model on the NTU-60, NTU-120 and PKU-II datasets. We present our results in Table 1. We see that in both uni-modal and multi-modal training our approach outperforms the state-of-the-art on this protocol which shows that our approach learns better features. Further, it is interesting to see that apart from outperforming all single modality baselines, training using HaLP on a single modality even shows competitive performance to models trained using multiple modalities.

kNN Evaluation: This is a hyperparameter-free & training-free evaluation protocol which directly uses the pre-trained encoder and applies a k-Nearest Neighbor classifier (k=1) to the learned features of the training samples. We present our results using this protocol in Table 2. We again see considerable improvements on the various task settings showing that our approach works better even in a hyperparameter-free evaluation setting.

Table 2. kNN evaluation: In addition to linear probing, we also show improved performances on kNN evaluation. Similar to linear probing our proposed module HaLP leads to significant performance improvements on both NTU-60 and NTU-120 datasets compared to the single-modality baseline. Using our approach with CMD [37] leads to further gains over state-of-the-art.

Method	NTU-60		NTU-120	
	x-sub	x-view	x-sub	x-set
Additional training modalities or encoders				
ISC [48]	62.5	82.6	50.6	52.3
CrosSCLR-B	66.1	81.3	52.5	54.9
CMD	70.6	85.4	58.3	60.9
HaLP+CMD	**71.0**	**86.4**	**59.4**	**61.9**
Additional training modalities or encoders				
LongT GAN [61]	39.1	48.1	31.5	35.5
P&C [47]	50.7	76.3	39.5	41.8
Baseline	63.6	82.8	51.7	55.3
HaLP	**65.8**	**83.6**	**55.8**	**59.0**

Table 3. NTU to PKU transfer: We adapt NTU pretrained models to the PKU-II dataset. We observe that our model improves in the transfer learning setup as well.

Method	To PKU-II	
	NTU-60	NTU-120
Additional training modalities or encoders		
ISC [48]	51.1	52.3
CrosSCLR-B	54.0	52.8
CMD	56.0	57.0
HaLP + CMD	**56.6**	**57.3**
Training using only joint		
LongT GAN [61]	44.8	-
MS^{2}L [36]	45.8	-
Baseline	53.3	53.4
HaLP	**54.8**	**55.4**

NTU to PKU transfer: Here, we evaluate whether the pre-trained models trained on NTU can be used to transfer to PKU, which has much less training data. A classifier MLP is attached to an NTU pre-trained encoder, and the entire model is finetuned with labels on the PKU dataset. In Table 3, we see that our approach improves over the state-of-the-art, showing more transferable features.

4.4. Analyses and ablations

Next, we present ablation analyses on our models. In this section, we work with the single-modality training of HaLP for NTU-60 Cross Subject split unless otherwise mentioned. Models for these experiments are pre-trained on NTU-60 using the cross-subject protocol, and results for

Table 4. Computational overhead: Our proposed module HaLP is lightweight and results in very small computational overheads and can potentially be added to any contrastive learning method.

Method	Time/epoch	Train GPU memory	NTU-60 x-sub
Baseline	1x	1x	78.0
HaLP	1.13x	1x	**79.7**
CMD	3x	1.94x	79.8
HaLP+CMD	3.32x	1.94x	**82.1**

Table 5. Effect of changing λ, which controls the maximum hardness of generated points. We find that that using a large value of λ (harder positives) shows better performance.

$\lambda \rightarrow$	1	0.8	0.6	0.4	0.2
NTU-60 x-sub	**79.7**	**79.7**	79.6	79.6	79.5

linear evaluation are presented.

Training time and Memory requirements: While our approach involves clustering on the hypersphere manifold, we empirically show that our approach incurs only marginal overheads compared to the baseline (Table 4). We see that HaLP, has performance comparable to CMD while incurring much less training memory and training time penalty, which makes use of multiple modalities during training. Further, HaLP+CMD is comparable to CMD in training time and memory requirements but outperforms CMD [37]. Note that since our technique modifies the loss, the inference pipeline remains exactly the same as the baseline, and we run at 1x the time of the baseline.

Can query-key be used to hallucinate positives? One potentially simple way to generate positives would be to interpolate between z_k and z_q along the geodesic joining them. We observe that this approach does not improve over the baseline and has a performance of 77.92%. This shows that naively selecting directions to explore may not lead to better training.

Ablation on λ: Recollect that in Sec. 3.4, we choose $t_c \sim \texttt{uniform}(0, \lambda t^*)$. The value of λ effectively controls the maximum hardness of the generated positives. In Table 5, we modify λ, and observe that a value of 1.0 and 0.8 works well. We found that the $\lambda = 0.8$ works consistently well across all datasets and use it for all our experiments. Instead of generating positives with a range of hardness, one could also just use the positive defined by $t_c = t^*$. We find this is suboptimal (79.4%). Thus, the use of only the hardest positives is not very effective.

How many prototypes to use? In our approach, prototypes $\mathcal{P}$ are obtained using a k-Means clustering algorithm on the hypersphere manifold. Hallucinated latent positives are obtained by stepping along a randomly chosen prototype. In this experiment, we vary the number of clusters and their effect on the performance. In Table 6, we see that using 20 prototypes is optimal for NTU-60 dataset.

Table 6. Effect of varying the number of prototypes (N) to be extracted. We observe that using 20 prototypes for NTU-60 leads to optimal performance.

# Prototypes (N) $\rightarrow$	10	20	40	60
NTU-60 x-sub	79.5	**79.7**	79.5	79.5

Which anchor to use? : In our work, we choose z_k as the anchor used to hallucinate positives. An alternative could be to use z_q instead. We found that this does not improve over the baseline and has linear evaluation performance of 77.9%. This might be due to the nature of the loss since it compares the similarity of the generated positive to z_q which might lead to trivial training signals. Based on this observation, we use z_k as an anchor for our experiments.

Supplementary material: Additional experiments and analyses including semi-supervised learning, multi-modal ensembles, the effect of top-K, HaLP applied to other frameworks and tasks, additional results using multi-modal training, and alternate variants are provided in the supplementary material.

5. Conclusion

We present an approach to hallucinate positives in the latent space for self-supervised representation learning of skeleton sequences. We define an objective function to generate hard latent positives. On-the-fly generation of positives requires that the process is fast and introduces minimal overheads. To that end, we propose relaxations to the original objective and derive closed-form solutions for the hard positive. The generated positives with varying amounts of hardness are then used within a contrastive learning framework. Our approach offers a fast alternative to hand-crafting new augmentations. We show that our approach leads to state-of-the-art performance on various standard benchmarks in self-supervised skeleton representation learning.

6. Acknowledgements

AS acknowledges support through a fellowship from JHU + Amazon Initiative for Interactive AI (AI2AI). AR and RC acknowledge support from an ONR MURI grant N00014-20-1-2787. SM and DJ were supported in part by the National Science Foundation under grant no. IIS-1910132 and IIS-2213335. AS thanks Yunyao Mao for help with questions regarding CMD.

References

[1] Geomstats https://geomstats.github.io/. 7

[2] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com. 7

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020. 2, 4

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 2, 4

[5] Yao-Jen Chang, Shu-Fang Chen, and Jun-Da Huang. A kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. *Research in developmental disabilities*, 32(6):2566–2570, 2011. 1

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 4

[8] Yuxiao Chen, Long Zhao, Jianbo Yuan, Yu Tian, Zhaoyang Xia, Shijie Geng, Ligong Han, and Dimitris N Metaxas. Hierarchically self-supervised transformer for human skeleton representation learning. In *European Conference on Computer Vision*, pages 185–202. Springer, 2022. 3

[9] Yi-Bin Cheng, Xipeng Chen, Junhong Chen, Pengxu Wei, Dongyu Zhang, and Liang Lin. Hierarchical transformer: Unsupervised representation learning for skeleton-based human action recognition. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021. 3, 6

[10] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7024–7033, 2018. 3

[11] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020. 2, 3

[12] Kellie Corona, Katie Osterdahl, Roderic Collins, and Anthony Hoogs. Meva: A large-scale multiview, multimodal video dataset for activity detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1060–1068, 2021. 1

[13] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2969–2978, 2022. 3

[14] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 1

[15] Songwei Ge, Shlok Mishra, Chun-Liang Li, Haohan Wang, and David Jacobs. Robust contrastive learning using negative samples with diminished semantics. *Advances in Neural Information Processing Systems*, 34:27356–27368, 2021. 2, 3

[16] Aritra Ghosh and Andrew S. Lan. Contrastive learning improves model robustness under label noise. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2697–2702, 2021. 2

[17] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2

[18] Tianyu Guo, Hong Liu, Zhan Chen, Mengyuan Liu, Tao Wang, and Runwei Ding. Contrastive learning from extremely augmented skeleton sequences for self-supervised action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 762–770, 2022. 3, 6

[19] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010. 2

[20] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 2

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 2

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2, 3, 4

[23] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020. 2

[24] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2):201–211, 1973. 1

[25] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*, 33:21798–21809, 2020. 3, 4

[26] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5492–5501, 2019. 1

[27] Łukasz Kidziński, Bryan Yang, Jennifer L Hicks, Apoorva Rajagopal, Scott L Delp, and Michael H Schwartz. Deep

neural networks enable quantitative movement analysis using single-camera videos. *Nature communications*, 11(1):1–10, 2020. 1

[28] Boeun Kim, Hyung Jin Chang, Jungho Kim, and Jin Young Choi. Global-local motion transformer for unsupervised skeleton-based action learning. In *European Conference on Computer Vision*, pages 209–225. Springer, 2022. 3, 6

[29] Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *International Journal of Computer Vision*, 130(5):1366–1401, 2022. 1

[30] Skanda Koppula, Yazhe Li, Evan Shelhamer, Andrew Jaegle, Nikhil Parthasarathy, Relja Arandjelović, João Carreira, and Olivier J. H'enaff. Where should i spend my flops? efficiency evaluations of visual pre-training methods. *ArXiv*, abs/2209.15589, 2022. 2

[31] Kibok Lee, Yian Zhu, Kihyuk Sohn, Chun-Liang Li, Jinwoo Shin, and Honglak Lee. i-mix: A domain-agnostic strategy for contrastive representation learning. *arXiv preprint arXiv:2010.08887*, 2020. 3

[32] Alexander Li, Alexei A. Efros, and Deepak Pathak. Understanding collapse in non-contrastive siamese representation learning. In *ECCV*, 2022. 2

[33] Linguo Li, Minsi Wang, Bingbing Ni, Hang Wang, Jiancheng Yang, and Wenjun Zhang. 3d human action representation learning via cross-view consistency pursuit. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4741–4750, 2021. 3

[34] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 513–528, 2018. 1

[35] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9572–9581, 2019. 1

[36] Lilang Lin, Sijie Song, Wenhan Yang, and Jiaying Liu. Ms2l: Multi-task self-supervised learning for skeleton based action recognition. *Proceedings of the 28th ACM International Conference on Multimedia*, 2020. 6, 7

[37] Yunyao Mao, Wengang Zhou, Zhenbo Lu, Jiajun Deng, and Houqiang Li. Cmd: Self-supervised 3d action representation learning with cross-modal mutual distillation. *arXiv preprint arXiv:2208.12448*, 2022. 2, 3, 6, 7, 8

[38] Yue Meng, Rameswar Panda, Chung-Ching Lin, Prasanna Sattigeri, Leonid Karlinsky, Kate Saenko, Aude Oliva, and Rogerio Feris. Adafuse: Adaptive temporal fusion network for efficient action recognition. *arXiv preprint arXiv:2102.05775*, 2021. 1

[39] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016. 2

[40] Qiang Nie, Ziwei Liu, and Yunhui Liu. Unsupervised 3d human pose representation with viewpoint and pose disentanglement. In *ECCV*, 2020. 6

[41] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 2

[42] Haocong Rao, Shihao Xu, Xiping Hu, Jun Cheng, and Bin Hu. Augmented skeleton based contrastive action learning with momentum lstm for unsupervised action recognition. *Information Sciences*, 569:90–109, 2021. 3, 6

[43] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020. 3, 4

[44] Anshul Shah, Suvrit Sra, Rama Chellappa, and Anoop Cherian. Max-margin contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8220–8230, 2022. 3, 4

[45] Zhiqiang Shen, Zechun Liu, Zhuang Liu, Marios Savvides, Trevor Darrell, and Eric Xing. Un-mix: Rethinking image mixtures for unsupervised visual representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2216–2224, 2022. 3

[46] Junhyuk So, Changdae Oh, Minchul Shin, and Kyungwoo Song. Multi-modal mixup for robust fine-tuning. *arXiv preprint arXiv:2203.03897*, 2022. 3

[47] Kun Su, Xiulong Liu, and Eli Shlizerman. Predict & cluster: Unsupervised skeleton based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9631–9640, 2020. 3, 6, 7

[48] Fida Mohammad Thoker, Hazel Doughty, and Cees GM Snoek. Skeleton-contrastive 3d action representation learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1655–1663, 2021. 2, 3, 6, 7

[49] James Townsend, Niklas Koep, and Sebastian Weichwald. Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *arXiv preprint arXiv:1603.03236*, 2016. 4

[50] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019. 3, 5

[51] Zekai Wang and Weiwei Liu. Robustness verification for contrastive learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 22865–22883. PMLR, 17–23 Jul 2022. 2

[52] Shih-En Wei, Nick C Tang, Yen-Yu Lin, Ming-Fang Weng, and Hong-Yuan Mark Liao. Skeleton-augmented human action understanding by learning with progressively refined data. In *Proceedings of the 1st ACM International Workshop on Human Centered Event Understanding from Multimedia*, pages 7–10, 2014. 1

[53] Philippe Weinzaepfel and Grégory Rogez. Mimetics: Towards understanding human actions out of context. *International Journal of Computer Vision*, 129(5):1675–1690, 2021. 1

[54] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018. 3

[55] Siyuan Yang, Jun Liu, Shijian Lu, Meng Hwa Er, and Alex C Kot. Skeleton cloud colorization for unsupervised 3d action representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13423–13433, 2021. 3

[56] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 3

[57] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 3

[58] Haoyuan Zhang, Yonghong Hou, and Wenjing Zhang. Skeletal twins: Unsupervised skeleton-based action representation learning. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022. 3, 6

[59] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 2

[60] Shaofeng Zhang, Meng Liu, Junchi Yan, Hengrui Zhang, Lingxiao Huang, Pinyan Lu, and Xiaokang Yang. m-mix: Generating hard negatives via multiple samples mixing for contrastive learning. 2021. 3

[61] Nenggan Zheng, Jun Wen, Risheng Liu, Liangqu Long, Jianhua Dai, and Zhefeng Gong. Unsupervised representation learning with long-term dynamics for skeleton based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 3, 6, 7