Action Representing by Constrained Conditional Mutual Information

Haoyuan Gao[1], Yifan Zhang[1,3], Linhui Sun[1,2], Jian Cheng[1,3]

[1] Institute of Automation, Chinese Academy of Sciences
{gaohaoyuan2019,sunlinhui2018}@ia.ac.cn
{yfzhang,jcheng}@nlpr.ia.ac.cn
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3] AIRIA

Abstract. Contrastive learning achieves a remarkable performance for representation learning by constructing the InfoNCE loss function. It enables learned representations to describe the invariance in data transformation without labels. Contrastive learning also been employed in self-supervised learning of action recognition. However, this kind of method fails to introduce assumptions according to human knowledge about the prior distribution of representations in the training process. For solving this problem, this paper proposes a self-supervised learning framework, which can achieve different self-supervised learning methods by choosing different assumptions about the prior distribution of representations, while still learning the description of invariance in data transformation as contrastive learning. This framework minimizes the CCMI (Constrained Conditional Mutual Information) loss function, which represents the conditional mutual information between input augmented samples of the same sample and the output representations of the encoder while the prior distribution of representations is constrained. By theoretical analysis of the framework, it is proved that traditional contrastive learning by InfoNCE is a special case without human knowledge constraint of this framework. The Gaussian Mixture Model on Unit Hyper-sphere is chosen as the representation prior distribution to achieve the self-supervised method called CoMInG. Compared with the existing methods, the performance of the learned representation by this method in the downstream task of action recognition is significantly improved.

1 Introduction

Action recognition is widely used in video surveillance, human-computer interaction, and video understanding. The methods of action recognition include RGB image-based, depth image-based and skeleton-based methods. Recently, skeleton-based methods have attracted increasing attention due to their lower computation consumption and higher robustness against viewpoint variations and noisy backgrounds than the methods using RGB images and depth images [22]. Various skeleton-based works were proposed and achieved significant performance in recent years [6] [29] [32]. However, most of them utilize the supervised learning paradigm to learn action representations, which require massive labeled samples for training. This leads to some problems, including the high cost of labeling and the risk of mislabeling due to the high inter-class similarity

of the actions. In addition, massive valuable information for learning implied in unlabeled data is not utilized in the training. Therefore, self-supervised learning of action representation attracted increasing attention.

Self-supervised learning is a type of unsupervised representation learning that creates learning target without human annotation to train the encoder network to obtain effective representation for downstream tasks. Only a few works focus on self-supervised learning of action recognition [21] [25] [33]. The best performer [21] of them used contrastive learning for action representing, constructing the InfoNCE loss function to enable learned representations to describe the invariance in data transformation. However, such contrastive learning methods fail to introduce assumptions according to human knowledge about the prior distribution of representations in the training process.

For solving this problem, this paper proposes a self-supervised learning framework, which can achieve different self-supervised learning methods by choosing different assumptions about the prior distribution of representations, while still learning the description of invariance in data transformation as contrastive learning. This framework minimizes the CCMI (Constrained Conditional Mutual Information) loss function, which represents the conditional mutual information between input augmented samples of the same sample and the output representations of the encoder while the prior distribution of representations is constrained. By theoretical analysis of the framework, it is proved that traditional contrastive learning by InfoNCE is a special case without human knowledge constraint of this framework. In this paper, the Gaussian Mixture Model on Unit Hyper-sphere is chosen as the representation prior distribution to introduce a self-supervised method named CoMInG (**Co**nstrained Conditional **M**utual **In**formation Minimizing with **G**aussian Mixture Model on Unit Hyper-sphere as Representation Prior), and employ it for action representing of skeleton-based action recognition.

Our contributions can be summarized as follows.

- This paper proposes a self-supervised learning framework by minimizing CCMI, which can achieve different self-supervised learning methods by choosing different assumptions about the prior distribution of representations, while still learning the description of invariance in data transformation as contrastive learning.
- By theoretical analysis of the framework, it is proved that traditional contrastive learning by InfoNCE is a special case without human knowledge constraint of this framework. This conclusion enhances the theoretical credibility of the proposed framework.
- This paper proposes a self-supervised learning method CoMInG, by assuming the prior of representations as a Gaussian Mixture Model on Unit Hyper-sphere, and comprehensively evaluates the effectiveness of CoMInG on three public datasets: NTU RGB+D 60, NTU RGB+D 120 and SBU datasets. Under the linear evaluation protocol, the proposed CoMInG achieves the best performance than existing self-supervised learning methods.

The rest of this paper is organized as follows: Section 2 introduces the previous works related to the proposed work. Section 3 introduces the details of CCMI loss and CoMInG method. Section 4 proves the relationship between InfoNCE and CCMI.

Section 5 compares CoMInG with existing baselines. Ablation studies are also provided in section 6. The conclusion of the proposed work is shown in Section 6.

2 Related Work

2.1 Self-Supervised Learning

Self-supervised learning aims to learn an effective representation from unlabeled data by using a self-supervised proxy task to train an encoder network. The learned representation can be transferred and beneficial to the downstream tasks [12].

Self-supervised learning methods include generative-based methods, contrastive-based methods, and clustering-based methods [18]. Generative-based methods employ an encoder-decoder structure or generative adversarial network to learn the representation. For example, [14] proposed a method to use an autoencoder as a generator with a discriminator for the automatic colorization of images. For the clustering-based methods, [2] presented DeepCluster, which iteratively groups the features with a standard clustering algorithm, K-means, and uses the subsequent assignments as supervision to update the weights of the encoder network. [1] proposed a method obtained by maximizing the information between labels and input data indices, using a fast variant of the Sinkhorn-Knopp algorithm. [3] proposed a framework both using the contrastive-based and clustering-based method, SwAV, which employs the Sinkhorn-Knopp algorithm to cluster the data and uses the cluster codeID of the other augmentations fromof the same image to guide the representations. PCL combined MoCo with an off-line k-means clustering process to propose the ProtoNCE loss [15]. Most of the state-of-the-art self-supervised learning methods are contrastive-based methods. Contrastive multiview coding (CMC) enforced the different views of the same image close to each other [26]. Momentum contrastive (MoCo) improved contrastive learning by introducing a momentum encoder and a queue-based memory bank [10]. Chen et al. proposed SimCLR, which adds a projector network behind the encoder and redesigns a stronger augmentation strategy for contrastive learning [4]. [8] introduced BYOL, which relies on two neural networks, referred to as online network and target network. It trains the online network to predict the target network representation of the same image under a different augmented view and updates the target network with a slow-moving average of the online network. Barlow Twins maximized the similarity between the correlation matrix of the output representations and the identity matrix, avoiding the complete collapse that is likely to appear in contrastive learning [31].

2.2 Action Recognition

Traditional skeleton-based methods recognized the pattern of action by designing hand-crafted descriptors, such as the method proposed by Oreifej et al. which learns features by a modified histogram of oriented gradients (HOG) algorithm [20]. Due to the significant development of deep learning, numerous methods for supervised skeleton-based action recognition by the deep neural network were proposed, such as Directed

Graph Neural Network (DGNN) proposed by Shi et al [24], DeCoupling Graph Convolutional Networks (DC-GCN) proposed by Cheng et al [5] and Spatial-temporal Graph Convolutional Networks (STGCN) proposed by Yan et al [29].

Only a few self-supervised methods were proposed in recent years. [33] proposed a method by both using an encoder-decoder structure and a generative adversarial structure to reconstruct a masked input sequence. [25] forced the encoder to learn the action representation by using an autoencoder to re-generate the skeleton sequence, and additionally proposed a decoder-weakening mechanism by fixing the decoder weights or decoder states. [16] proposed to integrate multiple self-supervised tasks, that are motion prediction, jigsaw puzzle recognition, and contrastive learning, to learn more general representations. [21] maximized the similarity between augmented instances of the same input skeleton sequence by a queue-based memory bank and momentum encoder. [28] proposed a self-supervised framework, which not only reconstructs sequence by an autoencoder but also regards the K-means clustering results of action representations as pseudo labels to train the encoder. Contrastive learning by InfoNCE has best performance in self-supervised action recoginition [21], but it cannot introduce assumptions according to human knowledge about the prior distribution of representations in the training process.

3 Methodology

3.1 Traditional Contrastive Learning by Minimizing InfoNCE

InfoNCE is a widely used loss function in self-supervised learning, which enables learned representations to describe the invariance in data transformation, that is:

$$L_{NCE} = -\frac{1}{I}\sum_{i=1}^{I}\log\frac{\exp\left(sim(z^{(i,1)}, z^{(i,2)})\right)}{\sum_{j=1}^{I}\exp\left(sim(z^{(i,1)}, z^{(j,2)})\right)} \tag{1}$$

I is the size of the dataset. $z^{(i,1)}$ and $z^{(i,2)}$ are respectively the representations of two randomly augmented version of ith sample extracted by the encoder network [4]. $sim(a, b)$ measures the similarity between two variables. The self-supervised methods using InfoNCE are called contrastive learning methods. Although contrastive learning achieves a remarkable performance for representation learning in multiple applications, it fails to introduce assumptions according to human knowledge about the prior distribution of representations in the training process. Therefore, this paper proposes the CCMI loss to solve this problem, and we will prove InfoNCE is a special case of CCMI in section 4.

3.2 A Learning Framework by Minimizing CCMI

This subsection proposes a simple framework for self-supervised learning by minimizing loss function called CCMI (Constrained Conditional Mutual Information). This framework can introduce various different self-supervised learning methods by different choices of representation prior according to human knowledge, while still learning the description of invariance in data transformation as contrastive learning.

Let us consider one dataset $X = \{x^i\}_{i=1}^I$ consisting of I i.i.d. samples of some continuous variable x, which is a skeleton sequence in our task. We assume that each sample is generated by a random process including an unobserved continuous random variable z. In addition, a variable v denotes the augmented sample for the sample x after some random data augmentation. Specifically, the generating process of x and v can be divided into three steps: (1) a value z^i is sampled from distribution $p_{\theta^*}(z)$; (2) a value x^i is sampled from distribution $p_{\theta^*}(x|z)$; (3) a value v^i is sampled from distribution $p_{\theta^*}(v|x,z)$. We assume that the $p_{\theta^*}(z)$, $p_{\theta^*}(x|z)$ and $p_{\theta^*}(v|x,z)$ come from parametric families of distributions $p_\theta(z)$, $p_\theta(x|z)$ and $p_\theta(v|x,z)$. Based on the used data augmentation strategy, we can reasonably assume that for any x^i and x^j ($i \neq j$), $v^i = v^j$ will never happen if v^i and v^j are respectively sampled from $p_{\theta^*}(v|x^i)$ and $p_{\theta^*}(v|x^j)$. Therefore, if $g(v)$ is a function that can find the only sample x corresponding to the augmented sample v, the conditional distribution $p_{\theta^*}(x|v)$ is $p_{\theta^*}(x|v) = 1\ if\ x = g(v)\ else\ 0$. Then we can obtain that $p_{\theta^*}(z|v) = \int p_{\theta^*}(x|v)p_{\theta^*}(z|x,v)dx = p_{\theta^*}(z|x,v)$.

Because we wish that the information of the representation vector z is only affected by the semantic information in the sample x that does not change with the data augmentation, the learning target is that when x is known, v and z have **conditional independence**, that is, $p_\theta(v|x,z) = p_\theta(v|x)$. Therefore, the learning target we set is:

$$
\begin{aligned}
\theta^* &= \arg\min_\theta KL(p_\theta(v|x,z)\|p_\theta(v|x)) \\
&= \arg\min_\theta \underset{p_\theta(z,x,v)}{E}[\log \frac{p_\theta(v|x,z)}{p_\theta(v|x)}] \\
&= \arg\min_\theta \underset{p_\theta(z,x,v)}{E}[\log \frac{p_\theta(v|x,z)p_\theta(z,x)}{p_\theta(v|x)p_\theta(z,x)}] \\
&= \arg\min_\theta \underset{p_\theta(z,x,v)}{E}[\log \frac{p_\theta(z|x,v)p_\theta(v|x)p_\theta(x)}{p_\theta(v|x)p_\theta(z|x)p(x)}] \\
&= \arg\min_\theta \underset{p_\theta(z,x,v)}{E}[\log \frac{p_\theta(z|v)p_\theta(v|x)}{p_\theta(v|x)p_\theta(z|x)}] \\
&= \arg\min_\theta \int p_\theta(x)[\iint p_\theta(z,v|x)\log \frac{p_\theta(z,v|x)}{p_\theta(v|x)p_\theta(z|x)}dzdv]dx \\
&= \arg\min_\theta I(Z;V|X)
\end{aligned}
\tag{2}
$$

where $I(Z;V|X)$ is the conditional mutual information between z and v, while x is known. When $I(Z;V|X)$ reaches its minimum value 0, $p_\theta(v|x,z) = p_\theta(v|x)$.

However, only minimizing $I(Z,V|X)$ can easily achieve the trivial solution, that is, whatever the value of v and x are, $p_\theta(z|v,x)$ is the same. A simple way to solve this problem is to inject our human knowledge into the hypothesis of prior distribution $p_\theta(z)$. We can choose a distribution $q_\phi(z)$ based on the human knowledge, and then set the optimization problem as:

$$
\theta^*, \phi^* = \arg\min_{\theta,\phi} I(Z;V|X) \quad s.t.\ KL(p_\theta(z)\|q_\phi(z)) = 0 \tag{3}
$$

Thanks to the constraint $KL(p_\theta(z)\|q_\phi(z)) = 0$, the trivial solution is avoided. For transform the optimization problem to a loss function for encoder training, we need to reform it. The Monte-Carlo Estimate [9] shows that:

$$\underset{p(x)}{E}[f(x)] = \int p(x)f(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(x^i) \tag{4}$$

where, x^i is sampled from $p(x)$.

Based on the Lagrangian multiplier method and Monte-Carlo Estimate, the above optimization problem is equal to:

$$\begin{aligned}
&\theta^*, \phi^*, \lambda^* = \arg\min_{\theta,\phi,\lambda} I(Z;V|X) + \lambda \underset{p_\theta(x,v|z)}{E}[KL(p_\theta(z)\|q_\phi(z))] \\
&= \arg\min_{\theta,\phi,\lambda} \iiint p_\theta(x)p_\theta(v|x)p_\theta(z|v)[\log\frac{p_\theta(z|v)}{p_\theta(z|x)} + \lambda\log\frac{p_\theta(z)}{q_\phi(z)}]dzdvdx \\
&= \arg\min_{\theta,\phi,\lambda} \iiint p_\theta(x)p_\theta(v|x)p_\theta(z|v)[\log p_\theta(z|v) - \log p_\theta(z|x) \\
&+ \lambda\log p_\theta(z) - \lambda\log q_\phi(z)] \\
&= \arg\min_{\theta,\phi,\lambda} \iiint p_\theta(x)p_\theta(v|x)p_\theta(z|v)[\log p_\theta(z|v) - \log(\int p_\theta(v|x)p_\theta(z|v)dv) \\
&+ \lambda\log(\iint p_\theta(x)p_\theta(v|x)p_\theta(z|v)dvdx) - \lambda\log q_\phi(z)]dzdvdx \\
&= \arg\min_{\theta,\phi,\lambda} \frac{1}{IML}\sum_{i=1}^{I}\sum_{m=1}^{M}\sum_{l=1}^{L}[-\log\sum_{n=1}^{N} p_\theta(z^{(i,m,l)}|v^{(i,n)}) \\
&+ \lambda\log\sum_{j=1}^{I}\sum_{n=1}^{N} p_\theta(z^{(i,m,l)}|v^{(j,n)}) - \lambda\log q_\phi(z^{(i,m,l)}) + \frac{1}{L}H(p_\theta(z|v^{(i,m)})) - \log N]
\end{aligned} \tag{5}$$

This loss function is called as CCMI (Constrained Conditional Mutual Information), where the $x^{(i)}$, $v^{(i,m)}$ and $z^{(i,m,l)}$ are sampled from $p_\theta(x)$, $p_\theta(v|x^{(i)})$ and $p_\theta(z|v^{(i,m)})$ respectively. I denotes the size of the dataset. Both M and N denote the number of times that sample v. L denotes the number of times that sample z. $H(p_\theta(z|v))$ denotes the entropy of $p_\theta(z|v)$. The illustration of the framework by minimizing CCMI loss is shown in Figure 1.

3.3 A Self-Supervised Method: CoMInG

In this subsection, we choose the Gaussian Mixture Model on Unit Hyper-sphere as the representation prior of CCMI to introduce a novel self-supervised method named CoMInG (**C**onstrained C**o**nditional **Mu**tual **I**nformation Mi**n**imizing with **G**aussian Mixture Model on Unit Hyper-sphere as Representation Prior).

For the training of encoder, the distribution $p_\theta(z|v)$ and $q_\phi(z)$ need to be determined. Following the paper of Variational Auto-Encoder [13], the $p_\theta(z|v)$ is set as:

$$p_\theta(z|v) = N(f_\eta(v), I) \tag{6}$$

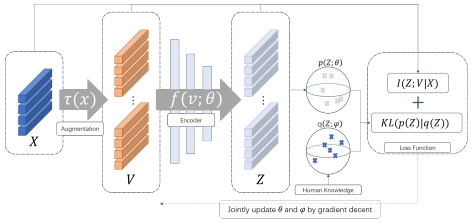

Fig. 1. Illustration of the proposed method.

where the f_η denotes our encoder network with the parameter η. The $N(\mu, S)$ denotes a multivariate Gaussian distribution with μ as the mean vector and S as the covariance matrix. In addition, we set the covariance matrix of the Gaussian distribution as identity matrix I. The reason of this setting is, in CCMI loss, the fourth part, the entropy of $p_\theta(z|v)$, needs to be minimized, which means the variance of Gaussian distribution needs to be minimized. For this target, we can primarily set the variance of Gaussian distribution as a small constant and we don't need to optimize this part in the training process.

The $q_\phi(z)$ is determined as a GMM (Gaussian Mixture Model) on Unit Hypersphere, due to the category information included in representation. We assume that each category match one Gaussian distribution in the GMM. We regard z as a sampling from one of the categories c, that is:

$$\begin{aligned} q_\phi(z) &= \sum_{c\in C} q(c)q(z|c) = \frac{1}{K}\sum_{k=1}^{K} N(w^k, I) \\ &= \frac{1}{K}\sum_{k=1}^{K} \frac{1}{2\pi^{\frac{D}{2}}} \exp\left(-\frac{(z-w^k)^{\mathrm{T}}(z-w^k)}{2}\right) \end{aligned} \tag{7}$$

We assume that for any k, $q(c^k) = 1/K$. I denotes the identity matrix, and D denotes the dimension of z. $W = \{w^k\}_{k=1}^K$ are K vectors that denote the mean vectors of the K Gaussian distributions. These mean vectors need to be optimized, which are same as the parameter η of the encoder network.

However, previous works prove that for contrastive learning, employing cosine similarity is better than using Euclidean distance in experiments [4], so we replace Euclidean distance in $q_\phi(z)$ with cosine similarity, that is:

$$q_\phi(z) = \sum_{c\in C} q(c)q(z|c) = \frac{1}{K}\sum_{k=1}^{K} \frac{1}{2\pi^{\frac{D}{2}}} \exp\left(\frac{sim(z, w^k)}{2}\right) \tag{8}$$

where $sim(z, w^k) = \frac{z^{\mathrm{T}} w^k}{\|z\|\|w^k\|}$.

However, due to the replacement, $\int q(z|c)dc = 1$ is no longer available. Therefore, $q(z|c)$ need to be normalized, that is:

$$q_\phi(z) = \sum_{c\in C} q(c)q(z|c) = \frac{1}{K}\sum_{k=1}^{K} \frac{1}{2\pi^{\frac{D}{2}} S} \exp\left(\frac{sim(z, w^k)}{2}\right) \tag{9}$$

where $S = \int \frac{1}{2\pi^{\frac{D}{2}}} \exp\left(\frac{sim(z,w^k)}{2}\right)dz$. This makes $\int q(z|c)dc = 1$ available again. Here Gaussian Mixture Model is transformed to the Gaussian Mixture Model on Unit Hyper-sphere. Other distributions can be chosen as $q_\phi(z)$ based on human knowledge, such as uniform distribution, exponential distribution, etc. According to the conclusion in f-gan [19], variational inference can be utilized to introduce all kinds of distribution as the target of prior.

According to the choice of $p_\theta(z|v)$ and $q_\phi(z)$, the loss function is:

$$\begin{aligned}
L_{CCMI} = \frac{1}{IML}\sum_{i=1}^{I}\sum_{m=1}^{M}\sum_{l=1}^{L}[&-\log\sum_{n=1}^{N} p_\theta(z^{(i,m,l)}|v^{(i,n)}) \\
&+ \lambda\log\sum_{j=1}^{I}\sum_{n=1}^{N} p_\theta(z^{(i,m,l)}|v^{(j,n)}) \\
&- \lambda\log q_\phi(z^{(i,m,l)}) + \frac{1}{L}H(p_\theta(z|v^{(i,m)})) - \log N] \\
= \frac{1}{IML}\sum_{i=1}^{I}\sum_{m=1}^{M}\sum_{l=1}^{L}[&-\log\sum_{n=1}^{N}\exp\left(sim(z^{(i,m,l)}, f_\eta(v^{(i,n)}))\right) \\
&+ \lambda\log\sum_{j=1}^{I}\sum_{n=1}^{N}\exp\left(sim(z^{(i,m,l)}, f_\eta(v^{(j,n)}))\right) \\
&- \lambda\log\sum_{k=1}^{K}\frac{1}{S}\exp\left(sim(z^{(i,m,l)}, w^{(k)})\right) + constant]
\end{aligned} \tag{10}$$

where D is the dimension of representations.

3.4 Details of Training Process

Following previous contrastive learning works [4] [10] and for fair comparison, in our experiments, we set $M = N = 1$. According to the result in the paper of Variational Auto-Encoder [13], we set $L = 1$. In addition, according to the ablation study, we set $\lambda = 1$. Based on this setting, when the constant term is omitted, the CCMI loss function becomes:

$$\begin{aligned}
L_{CCMI} = \frac{1}{I}\sum_{i=1}^{I}[&-sim(z^{(i,1)}, z^{(i,2)}) + \log\sum_{j=1}^{I}\exp\left(sim(z^{(i,1)}, z^{(j,2)})\right) \\
&- \log\sum_{k=1}^{K}\exp\left(sim(z^{(i,1)}, w^{(k)})\right)]
\end{aligned} \tag{11}$$

where $z^{(i,1)}$ is the representation vector of the augmented sample obtained by x after the first random data augmentation through the encoder, and $z^{(i,2)}$ is the representation vector of the augmented sample obtained by x after the second random data augmentation. Then the first and second parts of CCMI can be optimized directly by gradient

descent, but the third part of CCMI loss cannot. Thus we minimize the third part by EM algorithm.

The optimization problem of the third part of CCMI loss is:

$$\eta^*, \phi^* = \arg\min_{\eta,\phi} \log q_\phi(z) = \arg\min_{\eta,\phi} \log q_\phi(f_\eta(v)) \tag{12}$$

It is hard to optimize this function directly, so we use a surrogate function to higher-bound it:

$$\begin{aligned} \log q_\phi(f_\eta(v)) &= \log \sum_{c\in C} q_\phi(f_\eta(v), c) = \log \sum_{c\in C} q_\phi(c|f_\eta(v)) \frac{q_\phi(f_\eta(v), c)}{q_\phi(c|f_\eta(v))} \\ &\leq \sum_{c\in C} q_\phi(c|f_\eta(v)) \log \frac{q_\phi(f_\eta(v), c)}{q_\phi(c|f_\eta(v))} \end{aligned} \tag{13}$$

Then the E-step and M-step can be obtained:

– **E-Step**

$$q_{\phi^{(t)}}(c|f_{\eta^(t)}(v)) = q_{\phi^{(t)}}(c, f_{\eta^{(t)}}(v))/q_{\phi^{(t)}}(f_{\eta^{(t)}}(v)) \tag{14}$$

– **M-Step**

$$\eta^{(t+1)}, \phi^{(t+1)} = \arg\min_{\eta,\phi} \sum_{c\in C} q_{\phi^{(t)}}(c|f_{\eta^{(t)}}(v)) \log \left(q_\phi(f_\eta(v)|c)\right) \tag{15}$$

The pseudo-code of complete training process is shown in Algorithm 1. The parameter τ denotes a temperature parameter, which is widely used in previous contrastive learning methods to adjust the loss function [4]. The E-step is the calculation of γ^{ik} in the training process, and the M-step is the gradient descent in the training process.

4 Relationship between CCMI Loss and InfoNCE Loss

InfoNCE is a widely used loss function in previous self-supervised methods such as CPC, SimCLR and MoCo. This section proves that InfoNCE is a special case of CCMI.

In CoMInG, we set $p_\theta(z|v)$ as Gaussian distribution and set $\lambda = 1$, $M = N = 1$ and $L = 1$ for CCMI. If we change the choice of $q_\phi(z)$, and set that $q_\phi(z) = p_\theta(z|v)$, which means we have no human knowledge about prior distribution and set no limitation for it, then the CCMI loss function is:

$$\begin{aligned} L_{CCMI} = \frac{1}{IML} \sum_{i=1}^{I} \sum_{m=1}^{M} \sum_{l=1}^{L} [&- \log \sum_{n=1}^{N} P_\theta(z^{(i,m,l)}|v^{(i,n)}) \\ &+ \lambda \log \sum_{j=1}^{I} \sum_{n=1}^{N} P_\theta(z^{(i,m,l)}|v^{(j,n)}) - \lambda \log q_\phi(z^{(i,m,l)}) \\ &+ \frac{1}{L} H(p_\theta(z|v^{(i,m)})) - \log N] \end{aligned}$$

Algorithm 1 CoMInG

Input: initialized encoder parameters η, initialized mean vectors for Gaussian Mixture Model $W = \{w^k\}_{k=1}^K$, batch size B, learning rate α, temperature parameter τ, augmentation strategy T, similarity measurement $sim(a, b)$

for all minibatch in one epoch **do**
  Sample augmentation t^1 and t^2 from T
  for sampled minibatch $\{x^i\}_{i=1}^B$ **do**
    for sampled minibatch $\{x^j\}_{j=1}^B$ **do**
      $v^{(i,1)} = t^1(x^{(i)})$,$v^{(i,2)} = t^2(x^{(i)})$, $v^{(j,2)} = t^2(x^{(j)})$
      $z^{(i,1)} = f_\eta(v^{(i,1)})$,$z^{(i,2)} = f_\eta(v^{(i,2)})$,$z^{(j,2)} = f_\eta(v^{(j,2)})$
      for $k = 1 : K$ **do**
        $\gamma^{ik} = \frac{\exp(sim(z^{(i,1)}, w^{(k)}))}{\sum_{l=1}^K \exp(sim(z^{(i,2)}, w^{(l)}))}$
      end for
    end for
  end for
  $L_{ccmi} = -\frac{1}{B}[\sum_{i=1}^B sim(z^{(i,1)}, z^{(i,2)})/\tau$
    $+ \log \sum_{j=1}^B \exp(sim(z^{(i,1)}, z^{(j,2)})/\tau)]$
    $- \log \sum_{k=1}^K \gamma^{ik} \exp(sim(z^{(i,1)}, w^{(k)})/\tau)]$
  $W = W - \alpha \frac{\partial L_{ccmi}}{\partial W}$, $\eta = \eta - \alpha \frac{\partial L_{ccmi}}{\partial \eta}$
end for
return encoder parameterss η

$$
\begin{aligned}
&= \frac{1}{IML} \sum_{i=1}^{I} \sum_{m=1}^{M} \sum_{l=1}^{L} [-\log \sum_{n=1}^{N} P_\theta(z^{(i,m,l)}|v^{(i,n)}) \\
&+ \log \sum_{j=1}^{I} \sum_{n=1}^{N} P_\theta(z^{(i,m,l)}|v^{(j,n)}) - \frac{1}{L} \int p_\theta(z|v^{(i,m)}) \log p_\theta(z|v^{(i,m)}) dz \\
&+ \frac{1}{L} H(p_\theta(z|v^{(i,m)})) - \log N] \\
&= \frac{1}{IML} \sum_{i=1}^{I} \sum_{m=1}^{M} \sum_{l=1}^{L} [-\log \sum_{n=1}^{N} P_\theta(z^{(i,m,l)}|v^{(i,n)}) \\
&+ \log \sum_{j=1}^{I} \sum_{n=1}^{N} P_\theta(z^{(i,m,l)}|v^{(j,n)}) + constant] \\
&= \frac{1}{IML} \sum_{i=1}^{I} \sum_{m=1}^{M} \sum_{l=1}^{L} [-\log \sum_{n=1}^{N} \exp(sim(z^{(i,m,l)}, f_\eta(v^{(i,n)}))) \\
&+ \log \sum_{j=1}^{I} \sum_{n=1}^{N} \exp(sim(z^{(i,m,l)}, f_\eta(v^{(j,n)}))) + constant] \\
&= -\frac{1}{I} \sum_{i=1}^{I} \log \frac{\exp(sim(z^{(i,1)}, z^{(i,2)}))}{\sum_{j=1}^{I} \exp(sim(z^{(i,1)}, z^{(j,2)}))} + constant
\end{aligned} \tag{16}
$$

This loss function equals to InfoNCE, so that InfoNCE is a special case of CCMI when we set $\lambda = 1$, $M = N = 1$, $L = 1$, $p_\theta(z|v) = N(f_\eta(v), I)$ and $q_\phi(z) = p_\theta(z|v)$. This conclusion reveals that the essence of minimizing InfoNCE is to minimize the constrained conditional mutual information without injecting human knowledge to constrain the representation prior distribution.

5 Experiments

5.1 The setting of experiments

For evaluation, we conduct our experiments on commonly used three datasets: NTU60 dataset [23] (56578 samples, 60 categories), NTU120 dataset [17] (113945 samples, 120 categories) and SBU dataset [30] (282 samples, 8 categories).

In the experiments, the sequence length is set to 150, 150, and 40 for NTU60, NTU120, and SBU, respectively. The coordinate of the middle spine joint is subtracted by the coordinates of all joints for normalizing the skeleton sequences. Our encoder network adopts the LSTM with 512 hidden units.

In self-supervised pre-training, the encoder is trained by CoMInG. The batch size is 32, 32, and 128 for NTU60, NTU120, and SBU respectively. The network is trained by the SGD optimizer. The weight decay and momentum are set to 1e-4 and 0.9, respectively. We run the pre-training process for 60 epochs and the learning rate is multiplied by 0.1 per 30 epochs with 0.01 as the initialization. According to the results of ablation studies, we set the temperature τ as 0.06. The number of Gaussian of representation prior to 120, 150, and 30 for NTU60, NTU120, and SBU datasets respectively. The random data augmentations used in pre-training are 'Axis2Zero' and 'Shear'. For each joint in each sample, 'Axis2Zero' randomly chooses one of the axes of the 3D coordinates of the joint and changes it to zero, and the 'Shear' augmentation replaces each joint in a fixed direction [21].

After self-supervised pre-training, we use the linear evaluation to test our method. Specifically, the pre-trained encoder network by self-supervised learning is attached to a linear classifier, and we train the linear classifier for 90 epochs using skeleton sequences and labels in the training set while the parameter of the encoder is frozen. No augmentation is adopted in the training of the linear classifier. Then the Top-1 accuracy on the testing set is used to evaluate the effectiveness of the representations. The optimizer for training is stochastic gradient descent with a Nesterov momentum set as 0.9. The initialization of the learning rate is 1, and the learning rate decays at 15, 35, 60, and 75 epochs by 0.5.

5.2 Comparison with Existing Methods

Table 1, Table 2 and Table 3 compare the results of our approach with supervised approaches (using RNN as the backbone) and previous self-supervised approaches on SBU, NTU60 and NTU120 datasets in the linear evaluation setting respectively. The compared self-supervised approaches include all state-of-the-art approaches in this domain. "*" represents that we use the code shared by authors of original papers to obtain

ID	Method	Fold					
		1	2	3	4	5	Avg
	Supervised						
1	RNN	40.0	42.3	26.8	27.8	35.4	34.5
2	GRU	40.0	40.4	28.6	33.3	40.0	36.5
3	LSTM	49.1	53.2	37.5	42.0	53.8	47.1
	Self-supervised						
4	*P&C FW [25]	16.4	15.4	21.4	27.8	15.6	19.3
5	*PCRP [28]	16.4	36.5	21.4	24.1	29.7	25.6
6	ASCAL [21]	52.7	46.2	41.1	31.5	41.5	42.6
7	⋆InfoNCE	61.8	53.8	46.4	50.0	53.1	53.0
8	Ours	**65.5**	**63.5**	**51.8**	**61.1**	**53.1**	**59.0**

Table 1. Comparison with supervised, and self-supervised methods on SBU dataset. Bold numbers refer to the best performers.

these results, because in the original papers authors don't show the performance of the methods on these datasets. "⋆" represents that this method is coded by us, using the setting of our method. The results in Table 1 show that our method achieves significant improvement over previous self-supervised approaches and supervised baselines on all testing folds on SBU datasets. The results in Table 2 and Table 3 show that our method outperforms all self-supervised approaches and supervised approaches with RNN backbone on both cross-view and cross-subject settings of NTU60 and both cross-set and cross-subject settings of NTU120. It is worth noting that our method outperforms the traditional contrastive learning with InfoNCE on both SBU (ID=7) and NTU60 (ID=10) datasets.

5.3 Ablations

Hyperparameters. Table 4 shows the performance with different λ on three datasets. λ measures the weight of the constraint of $p_\theta(z)$ in the loss function. The training process will pay more attention on the $KL(p_\theta(z)\|q_\phi(z))$ and less attention on $I(Z;V|X)$ when λ becomes larger. The results show that it performs best when $\lambda = 1$, CoMInG achieves the best result. In Table 5, the results show that when τ set as 0.06, the best performance is obtained.

In Table 6, we show the performance with different batch size on three datasets. The best batch size of NTU60 and SBU is 32 and 128 respectively. An interesting phenomenon is, for specific training epochs (here is 60), for large-scale dataset NTU60, smaller batch size has a significant advantage over the larger ones, and for small-scale datasets SBU, larger batch size has a significant advantage over the smaller ones. This phenomenon was also shown in previous paper [21].

The setting of prior distribution. In CoMInG, the GMM (Gaussian Mixture Model) on unit hyper-sphere is chosen as $q_\phi(z)$. Here we try different numbers of Gaussian

ID	Method	CView Acc(%)	CSub Acc(%)
	Supervised		
1	Lie Group [27]	52.8	50.1
2	HBRNN [7]	64.0	59.0
3	Deep RNN [17]	64.1	56.3
	Self-Supervised		
4	*PCL [28]	53.7	\
5	LongT GAN [33]	48.1	39.1
6	P&C FW [25]	44.3	50.8
7	MS2L [16]	\	52.6
8	PCRP [28]	63.5	53.9
9	ASCAL [21]	63.6	58.0
10	⋆InfoNCE	61.0	56.9
11	Ours	**69.4**	**59.8**

Table 2. Comparison with supervised, and self-supervised methods on NTU60 dataset. Bold numbers refer to the best performers.

ID	Method	CSet Acc(%)	CSub Acc(%)
	Supervised		
1	Soft RNN [11]	44.9	36.3
2	PA LSTM [23]	26.3	25.5
	Self-Supervised		
3	P&C FW [25]	42.7	41.7
4	PCRP [28]	45.1	41.7
5	ASCAL [21]	49.2	48.3
6	Ours	**50.7**	**49.4**

Table 3. Comparison with supervised, and self-supervised methods on NTU120 dataset. Bold numbers refer to the best performers.

Dataset	λ			
Acc(%)	1	2	3	4
NTU60	**69.4**	65.6	57.1	42.8

Table 4. Performances for using different λ on two datasets for training 60 epochs. For SBU and NTU60, the results are for fold1 and cross-view setting respectively.

Dataset	τ			
Acc(%)	0.03	0.06	0.1	0.3
SBU	60.0	**65.5**	61.8	56.4
NTU60	67.3	**69.4**	63.6	55.8

Table 5. Performances for using different τ on two datasets for training 60 epochs. For SBU and NTU60, the results are for fold1 and cross-view setting respectively.

for our mixture model as the prior distribution. The choices of best performances are shown in Table 7.

Dataset	batch size			
Acc(%)	32	64	128	256
SBU	23.6	56.4	**65.5**	58.2
NTU60	**69.4**	67.2	65.1	62.8

Table 6. Performances for using different batch size on two datasets for training 60 epochs. For SBU and NTU60, the results are for fold1 and cross-view setting respectively.

SBU	Num of Gauss				
Acc(%)	10	20	30	40	50
Top1	61.8	61.8	**65.5**	60.0	60.0
NTU60	Num of Gauss				
Acc(%)	30	60	90	120	150
Top1	65.4	67.4	68.2	**69.4**	67.9
NTU120	Num of Gauss				
Acc(%)	90	120	150	180	210
Top1	48.8	50.2	**50.7**	49.8	49.6

Table 7. Performances for using different number of Gaussian on three datasets for training 60 epochs. For SBU, NTU60 and NTU120, the results are for fold1, cross-view setting and cross-set setting respectively.

6 Conclusion

We propose a framework for self-supervised learning by minimizing the constrained conditional mutual information between input augmented samples of the same sample and the output representations of the encode, which can achieve different self-supervised learning methods by choosing different assumptions about the prior distribution of representations, while still learning the description of invariance in data transformation as contrastive learning. Theoretical analysis shows that contrastive learning by InfoNCE is a special case of the proposed framework without human knowledge constraint. Based on this framework, we introduce a self-supervised method by choosing the Gaussian Mixture Model on Unit Hyper-sphere as the prior distribution of representations, and employ it for unsupervised action representing of skeleton-based action recognition. Experimental results of the proposed method show significant improvement on various commonly used datasets for action recognition.

Acknowledgements. This work was supported in part by NSFC 62273347, the National Key Research and Development Program of China (2020AAA0103402), Jiangsu Leading Technology Basic Research Project (BK20192004), and NSFC 61876182.

References

1. Asano, Y.M., Rupprecht, C., Vedaldi, A.: Self-labelling via simultaneous clustering and representation learning. arXiv preprint arXiv:1911.05371 (2019)

2. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 132–149 (2018)
3. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. arXiv preprint arXiv:2006.09882 (2020)
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
5. Cheng, K., Zhang, Y., Cao, C., Shi, L., Cheng, J., Lu, H.: Decoupling gcn with dropgraph module for skeleton-based action recognition. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16. pp. 536–553. Springer (2020)
6. Cheng, K., Zhang, Y., He, X., Chen, W., Cheng, J., Lu, H.: Skeleton-based action recognition with shift graph convolutional network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 183–192 (2020)
7. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1110–1118 (2015)
8. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., et al.: Bootstrap your own latent: A new approach to self-supervised learning. arXiv preprint arXiv:2006.07733 (2020)
9. Hammersley, J., Morton, K.: A new monte carlo technique: antithetic variates. In: Mathematical proceedings of the Cambridge philosophical society. vol. 52, pp. 449–475. Cambridge University Press (1956)
10. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9729–9738 (2020)
11. Hu, J.F., Zheng, W.S., Ma, L., Wang, G., Lai, J., Zhang, J.: Early action prediction by soft regression. IEEE transactions on pattern analysis and machine intelligence **41**(11), 2568–2583 (2018)
12. Jing, L., Tian, Y.: Self-supervised visual feature learning with deep neural networks: A survey. IEEE transactions on pattern analysis and machine intelligence (2020)
13. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
14. Larsson, G., Maire, M., Shakhnarovich, G.: Colorization as a proxy task for visual understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6874–6883 (2017)
15. Li, J., Zhou, P., Xiong, C., Hoi, S.C.: Prototypical contrastive learning of unsupervised representations. arXiv preprint arXiv:2005.04966 (2020)
16. Lin, L., Song, S., Yang, W., Liu, J.: Ms2l: Multi-task self-supervised learning for skeleton based action recognition. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 2490–2498 (2020)
17. Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.Y., Kot, A.C.: Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. IEEE transactions on pattern analysis and machine intelligence **42**(10), 2684–2701 (2019)
18. Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., Tang, J.: Self-supervised learning: Generative or contrastive. IEEE Transactions on Knowledge and Data Engineering (2021)
19. Nowozin, S., Cseke, B., Tomioka, R.: f-gan: Training generative neural samplers using variational divergence minimization. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. pp. 271–279 (2016)

20. Ohn-Bar, E., Trivedi, M.: Joint angles similarities and hog2 for action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 465–470 (2013)
21. Rao, H., Xu, S., Hu, X., Cheng, J., Hu, B.: Augmented skeleton based contrastive action learning with momentum lstm for unsupervised action recognition. Information Sciences **569**, 90–109 (2021)
22. Ren, B., Liu, M., Ding, R., Liu, H.: A survey on 3d skeleton-based action recognition using learning method. arXiv preprint arXiv:2002.05907 (2020)
23. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1010–1019 (2016)
24. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Skeleton-based action recognition with directed graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7912–7921 (2019)
25. Su, K., Liu, X., Shlizerman, E.: Predict & cluster: Unsupervised skeleton based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9631–9640 (2020)
26. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16. pp. 776–794. Springer (2020)
27. Vemulapalli, R., Arrate, F., Chellappa, R.: Human action recognition by representing 3d skeletons as points in a lie group. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 588–595 (2014)
28. Xu, S., Rao, H., Hu, X., Hu, B.: Prototypical contrast and reverse prediction: Unsupervised skeleton based action recognition. arXiv preprint arXiv:2011.07236 (2020)
29. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-second AAAI conference on artificial intelligence (2018)
30. Yun, K., Honorio, J., Chattopadhyay, D., Berg, T.L., Samaras, D.: Two-person interaction detection using body-pose features and multiple instance learning. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. pp. 28–35. IEEE (2012)
31. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. In: International Conference on Machine Learning. pp. 12310–12320. PMLR (2021)
32. Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N.: View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2117–2126 (2017)
33. Zheng, N., Wen, J., Liu, R., Long, L., Dai, J., Gong, Z.: Unsupervised representation learning with long-term dynamics for skeleton based action recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)