Learning View-Disentangled Human Pose Representation by Contrastive Cross-View Mutual Information Maximization: Supplementary Materials

Long Zhao[1,*] Yuxiao Wang[2] Jiaping Zhao[2] Liangzhe Yuan[2] Jennifer J. Sun[3]
Florian Schroff[2] Hartwig Adam[2] Xi Peng[4] Dimitris Metaxas[1] Ting Liu[2]

[1]Rutgers University [2]Google Research [3]Caltech [4]University of Delaware

1. Theoretical Results

1.1. Proofs

This section provides detailed proofs on how Eq. (2) is simplified to Eq. (5) in the main paper. We begin by introducing the following two propositions on MI.

Proposition 1. *Given any three random variables $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$, with a joint distribution $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. If $\boldsymbol{y}$ and $\boldsymbol{z}$ are independent and conditionally independent, i.e., $p(\boldsymbol{y}, \boldsymbol{z}) = p(\boldsymbol{y})p(\boldsymbol{z})$ and $p(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})$, we have that,*

$$\mathcal{I}(\boldsymbol{x}; \boldsymbol{y}, \boldsymbol{z}) = \mathcal{I}(\boldsymbol{x}; \boldsymbol{y}) + \mathcal{I}(\boldsymbol{x}; \boldsymbol{z}).$$

Proof. We start by applying the chain rule to $\mathcal{I}(\boldsymbol{x}; \boldsymbol{y}, \boldsymbol{z})$:

$$\mathcal{I}(\boldsymbol{x}; \boldsymbol{y}, \boldsymbol{z}) = \mathcal{I}(\boldsymbol{x}; \boldsymbol{y}|\boldsymbol{z}) + \mathcal{I}(\boldsymbol{x}; \boldsymbol{z}).$$

Then we have,

$$\begin{aligned}\mathcal{I}(\boldsymbol{x}; \boldsymbol{y}|\boldsymbol{z}) &= \sum_{\boldsymbol{z}}\sum_{\boldsymbol{y}}\sum_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \log \frac{p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})p(\boldsymbol{z})}{p(\boldsymbol{x}, \boldsymbol{z})p(\boldsymbol{y}, \boldsymbol{z})} \\ &= \sum_{\boldsymbol{z}}\sum_{\boldsymbol{y}}\sum_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \log \frac{p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z})}{p(\boldsymbol{x}|\boldsymbol{z})},\end{aligned}$$

and,

$$\begin{aligned}\mathcal{I}(\boldsymbol{x}; \boldsymbol{z}) &= \sum_{\boldsymbol{z}}\sum_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{z}) \log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{p(\boldsymbol{x})p(\boldsymbol{z})} \\ &= \sum_{\boldsymbol{z}}\sum_{\boldsymbol{x}} p(\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x}) \log \frac{p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p(\boldsymbol{x})p(\boldsymbol{z})} \\ &= \sum_{\boldsymbol{z}}\sum_{\boldsymbol{x}} p(\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x}) \log \frac{p(\boldsymbol{x}|\boldsymbol{z})}{p(\boldsymbol{x})} \\ &= \sum_{\boldsymbol{z}}\sum_{\boldsymbol{y}}\sum_{\boldsymbol{x}} p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x}) \log \frac{p(\boldsymbol{x}|\boldsymbol{z})}{p(\boldsymbol{x})} \\ &= \sum_{\boldsymbol{z}}\sum_{\boldsymbol{y}}\sum_{\boldsymbol{x}} p(\boldsymbol{x})p(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}) \log \frac{p(\boldsymbol{x}|\boldsymbol{z})}{p(\boldsymbol{x})} \\ &= \sum_{\boldsymbol{z}}\sum_{\boldsymbol{y}}\sum_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \log \frac{p(\boldsymbol{x}|\boldsymbol{z})}{p(\boldsymbol{x})}.\end{aligned}$$

After combining them together, we can show that,

$$\begin{aligned}\mathcal{I}(\boldsymbol{x}; \boldsymbol{y}, \boldsymbol{z}) &= \sum_{\boldsymbol{z}}\sum_{\boldsymbol{y}}\sum_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \log \frac{p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z})}{p(\boldsymbol{x})} \\ &= \sum_{\boldsymbol{z}}\sum_{\boldsymbol{y}}\sum_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \log \frac{p(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{y}, \boldsymbol{z})} \\ &= \sum_{\boldsymbol{z}}\sum_{\boldsymbol{y}}\sum_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \log \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{y})p(\boldsymbol{z})} \\ &= \sum_{\boldsymbol{z}}\sum_{\boldsymbol{y}}\sum_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \log \left[\frac{p(\boldsymbol{y}|\boldsymbol{x})}{p(\boldsymbol{y})} \frac{p(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z})}\right].\end{aligned}$$

Hence, we can rewrite $\mathcal{I}(\boldsymbol{x}; \boldsymbol{y}, \boldsymbol{z})$ as:

$$\begin{aligned}\mathcal{I}(\boldsymbol{x}; \boldsymbol{y}, \boldsymbol{z}) = &\sum_{\boldsymbol{z}}\sum_{\boldsymbol{y}}\sum_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \log \frac{p(\boldsymbol{y}|\boldsymbol{x})}{p(\boldsymbol{y})} \\ &+ \sum_{\boldsymbol{z}}\sum_{\boldsymbol{y}}\sum_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \log \frac{p(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z})}.\end{aligned}$$

[*]This work was done while the author was a research intern at Google.

According to the definition of MI, we have that,

$$\begin{aligned}
&\sum_{\boldsymbol{z}}\sum_{\boldsymbol{y}}\sum_{\boldsymbol{x}} p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})\log\frac{p(\boldsymbol{y}|\boldsymbol{x})}{p(\boldsymbol{y})}\\
=&\sum_{\boldsymbol{z}}\sum_{\boldsymbol{y}}\sum_{\boldsymbol{x}} p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})\log\frac{p(\boldsymbol{y}|\boldsymbol{x})}{p(\boldsymbol{y})}\\
=&\sum_{\boldsymbol{y}}\sum_{\boldsymbol{x}} p(\boldsymbol{x},\boldsymbol{y})\log\frac{p(\boldsymbol{y}|\boldsymbol{x})}{p(\boldsymbol{y})}=\mathcal{I}(\boldsymbol{x};\boldsymbol{y}),
\end{aligned}$$

and,

$$\begin{aligned}
&\sum_{\boldsymbol{z}}\sum_{\boldsymbol{y}}\sum_{\boldsymbol{x}} p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})\log\frac{p(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z})}\\
=&\sum_{\boldsymbol{z}}\sum_{\boldsymbol{y}}\sum_{\boldsymbol{x}} p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})\log\frac{p(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z})}\\
=&\sum_{\boldsymbol{z}}\sum_{\boldsymbol{x}} p(\boldsymbol{x},\boldsymbol{z})\log\frac{p(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z})}=\mathcal{I}(\boldsymbol{x};\boldsymbol{z}),
\end{aligned}$$

which prove the proposition. □

The next proposition is a minor adaptation of [20] according to Data Processing Inequality (DPI) [3].

Proposition 2 (Shwartz & Tishby [20])**.** *Given any two random variables $\boldsymbol{x}$ and $\boldsymbol{y}$, and any representation variable $\boldsymbol{z}$, defined as a (possibly stochastic) map of the input $\boldsymbol{x}$, we have the following DPI chain:*

$$\mathcal{I}(\boldsymbol{x};\boldsymbol{y})\geqslant\mathcal{I}(\boldsymbol{z};\boldsymbol{y}).$$

As defined in the main paper, we let $\boldsymbol{x}^i$ denote the given 2D pose from the i-th view. We are interested in learning an encoding network E that produces a view representation $\boldsymbol{z}_v^i$, and a pose representation $\boldsymbol{z}_p^i$ from the input $\boldsymbol{x}^i$. For simplicity, we define an optimal intermediate representation $\boldsymbol{z}^i$ that satisfies $p(\cdot,\boldsymbol{z}^i)=p(\cdot,\boldsymbol{z}_p^i,\boldsymbol{z}_v^i)$, and we have $\mathcal{I}(\cdot;\boldsymbol{z}^i)=\mathcal{I}(\cdot;\boldsymbol{z}_p^i,\boldsymbol{z}_v^i)$. Therefore, based on Proposition 1, the following equation holds:

$$\mathcal{I}(\boldsymbol{x}^i;\boldsymbol{z}_p^j,\boldsymbol{z}_v^i)=\mathcal{I}(\boldsymbol{x}^i;\boldsymbol{z}_p^j)+\mathcal{I}(\boldsymbol{x}^i;\boldsymbol{z}_v^i).$$

According to Proposition 2, we have that,

$$\mathcal{I}(\boldsymbol{x}^i;\boldsymbol{z}_p^j)\geqslant\mathcal{I}(\boldsymbol{z}^i;\boldsymbol{z}_p^j)=\mathcal{I}(\boldsymbol{z}_p^i,\boldsymbol{z}_v^i;\boldsymbol{z}_p^j).$$

After applying Proposition 1, we have that,

$$\mathcal{I}(\boldsymbol{z}_p^i,\boldsymbol{z}_v^i;\boldsymbol{z}_p^j)=\mathcal{I}(\boldsymbol{z}_p^i;\boldsymbol{z}_p^j)+\mathcal{I}(\boldsymbol{z}_v^i;\boldsymbol{z}_p^j).$$

As $\boldsymbol{z}_v^i$ and $\boldsymbol{z}_p^j$ are independent (mutually exclusive), it holds that $\mathcal{I}(\boldsymbol{z}_v^i;\boldsymbol{z}_p^j)=0$. Therefore, we have that,

$$\mathcal{I}(\boldsymbol{x}^i;\boldsymbol{z}_p^j)\geqslant\mathcal{I}(\boldsymbol{z}_p^i,\boldsymbol{z}_v^i;\boldsymbol{z}_p^j)=\mathcal{I}(\boldsymbol{z}_p^i;\boldsymbol{z}_p^j).$$

Similarly, we have that,

$$\begin{aligned}
\mathcal{I}(\boldsymbol{x}^i;\boldsymbol{z}_v^i)&\geqslant\mathcal{I}(\boldsymbol{z}^i;\boldsymbol{z}_v^i)=\mathcal{I}(\boldsymbol{z}_p^i,\boldsymbol{z}_v^i;\boldsymbol{z}_v^i)\\
&=\mathcal{I}(\boldsymbol{z}_p^i;\boldsymbol{z}_v^i)+\mathcal{I}(\boldsymbol{z}_v^i;\boldsymbol{z}_v^i)\\
&=\mathcal{I}(\boldsymbol{z}_p^i;\boldsymbol{z}_v^i)+\mathcal{I}(\boldsymbol{z}_v^i;\boldsymbol{z}_v^i)\\
&=\mathcal{H}(\boldsymbol{z}_v^i),
\end{aligned}$$

where $\mathcal{H}$ is the Shannon entropy which is always non-negative, *i.e.*, $\mathcal{H}\geqslant 0$. Then, we have that,

$$\mathcal{I}(\boldsymbol{x}^i;\boldsymbol{z}_p^j,\boldsymbol{z}_v^i)\geqslant\mathcal{I}(\boldsymbol{z}_p^i;\boldsymbol{z}_p^j)+\mathcal{H}(\boldsymbol{z}_v^i)\geqslant\mathcal{I}(\boldsymbol{z}_p^i;\boldsymbol{z}_p^j).$$

Finally, we can obtain that,

$$\begin{aligned}
&\sum_i\mathcal{I}(\boldsymbol{x}^i;\boldsymbol{z}_p^i,\boldsymbol{z}_v^i)+\sum_{i\neq j}\mathcal{I}(\boldsymbol{x}^i;\boldsymbol{z}_p^j,\boldsymbol{z}_v^i)\\
\geqslant&\sum_i\mathcal{I}(\boldsymbol{x}^i;\boldsymbol{z}_p^i,\boldsymbol{z}_v^i)+\sum_{i\neq j}\mathcal{I}(\boldsymbol{z}_p^i;\boldsymbol{z}_p^j),
\end{aligned}$$

which demonstrates the claim.

1.2. Relation to Cross Reconstruction

In this section, we provide an intuitive explanation on the relationship between the proposed cross-view MI maximization and the conventional methods based on cross reconstruction [14, 16, 17, 18]. According to [7], in the context of representation learning, given the input $\boldsymbol{x}$ and its representation $\boldsymbol{z}$ encoded by a neural network, the reconstruction error can be related to the MI as follows:

$$\begin{aligned}
\mathcal{I}(\boldsymbol{x};\boldsymbol{z})&=\mathcal{H}(\boldsymbol{x})-\mathcal{H}(\boldsymbol{x}|\boldsymbol{z})\geqslant-\mathcal{H}(\boldsymbol{x}|\boldsymbol{z})\\
&=\sum_{\boldsymbol{z}}\sum_{\boldsymbol{x}} p(\boldsymbol{x},\boldsymbol{z})\log p(\boldsymbol{x}|\boldsymbol{z})\\
&=\sum_{\boldsymbol{z}}\sum_{\boldsymbol{x}} p(\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})\log p(\boldsymbol{x}|\boldsymbol{z})\\
&\geqslant\sum_{\boldsymbol{x}} p(\boldsymbol{x})\sum_{\boldsymbol{z}} p(\boldsymbol{z}|\boldsymbol{x})\log p(\boldsymbol{x}|\boldsymbol{z})\\
&=\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left\{\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}|\boldsymbol{x})}\left[\log p(\boldsymbol{x}|\boldsymbol{z})\right]\right\},
\end{aligned}$$

where the inequality in the second-to-last line is achieved according to Jensen's inequality if we assume that the probability density functions are convex.

In typical settings of reconstruction, $p(\boldsymbol{z}|\boldsymbol{x})$ can be interpreted as an encoder while $p(\boldsymbol{x}|\boldsymbol{z})$ is the decoder. For example, the Variational Auto-Encoders (VAEs) [11] approximate $p(\boldsymbol{z}|\boldsymbol{x})$ by a tractable variational distribution $q(\boldsymbol{z}|\boldsymbol{x})$ with the KL divergence term $\mathbb{D}_{\mathrm{KL}}[q(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z})]$ which ensures that the learned distribution q is similar to the true prior distribution. To maximize the above objective, reconstruction-type methods usually minimize the mean squared error between the input and reconstruction if a Gaussian distribution is assumed or binary cross-entropy

loss if a Bernoulli distribution is assumed. Therefore, intuitively, the reconstruction-type objective is a lower bound of MI, and similar conclusions exist for other generative models based on reconstruction [1, 6].

We can easily extend the above formulation to the proposed cross-view MI maximization setup in the main paper, where we have that,

$$\begin{aligned}
&\sum_i \mathcal{I}(\boldsymbol{x}^i; \boldsymbol{z}_p^i, \boldsymbol{z}_v^i) + \sum_{i \neq j} \mathcal{I}(\boldsymbol{x}^i; \boldsymbol{z}_p^j, \boldsymbol{z}_v^i) \\
\geqslant &\sum_i \mathbb{E}_{p(\boldsymbol{x}^i)} \left\{ \mathbb{E}_{p(\boldsymbol{z}_p^i, \boldsymbol{z}_v^i | \boldsymbol{x}^i)} \left[\log p(\boldsymbol{x}^i | \boldsymbol{z}_p^i, \boldsymbol{z}_v^i) \right] \right\} \\
&+ \sum_{i \neq j} \mathbb{E}_{p(\boldsymbol{x}^i, \boldsymbol{x}^j)} \left\{ \mathbb{E}_{p(\boldsymbol{z}_p^j, \boldsymbol{z}_v^i | \boldsymbol{x}^i, \boldsymbol{x}^j)} \left[\log p(\boldsymbol{x}^i | \boldsymbol{z}_p^j, \boldsymbol{z}_v^i) \right] \right\},
\end{aligned}$$

where we can see that in an approximate sense, the existing methods for view-disentangled representation learning [14, 16, 17, 18] based on cross-reconstruction maximize a lower bound of the proposed cross-view MI maximization.

2. Implementation Details

2.1. Representation Learning

The backbone network architecture for our encoding network E is based on [12]. We use two residual blocks, batch normalization, 0.25 dropout, and no maximum weight norm constraint [12]. Both the likelihood estimation network Q and discriminator D are implemented by multi-layer perceptrons (MLPs). To be specific, Q consists of two fully-connected layers where the first layer is followed by the batch normalization and ELU activation [2]; D contains three fully-connected layers where the first two layers are followed by the ReLU activation. All these networks are trained using AdaGrad [4] with a fixed learning rate of 0.02 for optimization. For fair comparisons, all the representation learning methods compared to in the experiments also use the same architecture and training setup as described here. We also note that the learned representations are fixed during downstream training.

2.2. Action Recognition

Penn Action. We use a simple temporal convolution network to extract temporal features from per-frame pose representations generated by the encoding network. Table 1 presents the architecture of this network. We use Adam [10] with a fixed learning rate of 1.0×10^{-5} for optimization. We set the size of mini-batches to 64 and the network is trained for 1×10^6 iterations. During network training, we perform data augmentation by randomly horizontally flipping all frames in a video.

NTU-RGB+D. We use ResNet1D which is a modified version of [5] as the backbone network for action recognition on this dataset. Its detailed network architecture is shown in Table 2. We use Adam [10] with a fixed learning rate of 1.0×10^{-3} for training the network. We set the size of mini-batches to 64 and the network is optimized for 1×10^6 iterations. Following the practice of [22], no data augmentation is performed during training on this dataset. We only use single-person action categories, including action labels from A1 to A49, in this experiment.

Layer	Output Size	Setting
Conv1D	166×64	1×7, stride 2, BN, ReLU, 0.5
Conv1D	83×128	1×7, stride 2, BN, ReLU, 0.5
Conv1D	42×256	1×7, stride 2, BN, ReLU, 0.5
Pooling	256	global average pooling
Dense	14	SoftMax

Table 1. Architecture used on Penn Action [23]. The setting of "1×7, stride 2, BN, ReLU, 0.5" refers to 1D Convolution with kernel size of 1×7 and stride 2 followed by batch normalization (BN), ReLU activation, and a dropout layer with 0.5 drop rate.

Layer	Output Size	Setting
Conv1D	300×64	1×7, stride 1, BN, ReLU
R-Conv1D		1×7, stride 2, BN, ReLU, 0.5
		1×5, stride 1, BN, ReLU, 0.5
		1×3, stride 1, BN
Shortcut	150×64	1×1, stride 2, BN, ReLU
R-Conv1D		1×7, stride 2, BN, ReLU, 0.5
		1×5, stride 1, BN, ReLU, 0.5
		1×3, stride 1, BN
Shortcut	75×128	1×1, stride 2, BN, ReLU
R-Conv1D		1×7, stride 1, BN, ReLU, 0.5
		1×5, stride 1, BN, ReLU, 0.5
		1×3, stride 1, BN
Shortcut	75×256	1×1, stride 1, BN, ReLU
Pooling	256	global average pooling
Dense	49	SoftMax

Table 2. Architecture used on NTU-RGB+D [19]. R-Conv1D represents the residual layer introduced in [5]. The setting of "1×7, stride 2, BN, ReLU, 0.5" refers to 1D Convolution with kernel size of 1×7 and stride 2 followed by batch normalization (BN), ReLU activation, and a dropout layer with 0.5 drop rate.

3. Additional Results

Effectiveness of Camera Augmentation. We evaluate the effectiveness of camera augmentation by training a variant of CV-MIM where camera augmentation is not utilized. As shown in Table 3, training with camera augmentation leads to around 1% and 2% performance improvements on Penn Action and NTU-RGB+D, respectively.

More Results on NTU-RGB+D. We report the classification accuracy and standard deviation of different models

Methods	Penn Action [23]	NTU-RGB+D [19]
CV-MIM	**91.75 ± 0.24**	**56.50 ± 0.13**
CV-MIM w/o CA	90.75 ± 0.30	54.50 ± 0.22

Table 3. Classification accuracy (%) and standard deviation of CV-MIM with or without camera augmentation (CA) on Penn Action [23] and NTU-RGB+D [19] with the setting of single-shot cross-view action recognition.

on NTU-RGB+D with the setting of single-shot cross-view action recognition averaged over five repeated runs. Table 4 shows the results. We can see that our method achieves the best mean accuracy and smallest standard deviations.

More Results with Limited-Supervision. We provide additional comparisons with view-disentangled representation learning baselines under limited-supervision. Results on Penn Action [23] and NTU-RGB+D [19] are reported in Tables 5 and 6, respectively. We can see that the proposed CV-MIM outperforms other methods by a large margin consistently under different ratios of training data.

View Classification. We explore the utility of learned view representations by applying them to the task of view classification on Penn Action [23]. In this experiment, our target is to predict the view category, *i.e.*, left, right, front, or back, for each frame in a video. This is achieved by training a linear classifier which takes the learned view representations as input and it is trained by the ground truth view labels provided by this dataset. We use AdaGrad [4] with a fixed learning rate of 1.0×10^{-2} for training the classifier. We set the size of mini-batches to 64 and the classifier is optimized for 1×10^4 iterations. The data split of the fully-supervised setting is utilized to train and evaluate all view-disentangled representation learning approaches. We also compare our method to a baseline which directly takes raw 2D poses as input.

Table 7 shows the results of each method. We observe that our model obtains the best performance among representation learning methods, and we also outperform the baseline taking 2D poses. These results demonstrate that our learned view representations manage to encode effective view information for 2D poses, and can serve as a strong model for view-relevant downstream tasks.

More Visual Results. We show more qualitative results when using the learned representations of our model for nearest neighbor retrieval on Human3.6M [8] in Fig. 1. We also show additional nearest neighbor retrieval results when applying our model on MPI-INF-3DHP [13] in Fig. 2. Interestingly, we find that our learned representations are able to generalize to new views and new poses contained in MPI-INF-3DHP.

References

[1] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *Int. Conf. Learn. Represent.*, 2017. 3, 5, 6

[2] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *Int. Conf. Learn. Represent.*, 2016. 3

[3] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012. 2

[4] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7):2121–2159, 2011. 3, 4

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 3

[6] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Int. Conf. Learn. Represent.*, 2017. 3, 5, 6

[7] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *Int. Conf. Learn. Represent.*, 2019. 2, 5, 6

[8] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(7):1325–1339, 2013. 4, 6

[9] Tae Soo Kim and Austin Reiter. Interpretable 3D human action analysis with temporal convolutional networks. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 1623–1631, 2017. 5, 6

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Int. Conf. Learn. Represent.*, 2014. 3

[11] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Int. Conf. Learn. Represent.*, 2014. 2, 5, 6

[12] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3D human pose estimation. In *Int. Conf. Comput. Vis.*, pages 2640–2649, 2017. 3

[13] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3D human pose estimation in the wild using improved CNN supervision. In *3DV*, pages 506–516, 2017. 4, 7

[14] Qiang Nie, Ziwei Liu, and Yunhui Liu. Unsupervised 3D human pose representation with viewpoint and pose disentanglement. In *Eur. Conf. Comput. Vis.*, 2020. 2, 3

[15] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 5, 6

Methods	VD	C1-R1	C1-R2	C2-R1	C2-R2	C3-R1	C3-R2	Average
Res-TCN [9]		40.78 ± 0.39	39.71 ± 0.66	30.02 ± 0.63	48.45 ± 0.37	48.90 ± 0.42	29.25 ± 0.37	39.52 ± 0.23
Auto-Encoder	✓	42.76 ± 0.93	40.79 ± 0.70	32.13 ± 1.77	44.86 ± 0.59	42.39 ± 2.60	32.38 ± 1.34	39.22 ± 0.58
VAE [11]	✓	49.96 ± 0.61	50.92 ± 0.61	38.50 ± 0.66	53.63 ± 0.30	54.40 ± 0.49	36.94 ± 0.50	47.39 ± 0.26
β-VAE [1, 6]	✓	49.06 ± 1.07	48.81 ± 0.67	40.58 ± 0.84	51.39 ± 0.76	52.09 ± 0.44	36.87 ± 0.60	46.47 ± 0.21
InfoNCE [15]	✓	43.11 ± 0.32	42.60 ± 0.35	35.11 ± 0.88	47.01 ± 0.47	48.15 ± 0.69	34.37 ± 1.65	41.72 ± 0.37
DIM [7]	✓	41.71 ± 0.43	42.03 ± 0.80	32.63 ± 0.28	44.65 ± 0.70	43.75 ± 0.37	33.19 ± 0.51	39.66 ± 0.22
Pr-VIPE [21]		56.28 ± 0.22	56.95 ± 0.58	50.09 ± 0.73	50.38 ± 0.39	57.03 ± 0.32	**55.41 ± 0.38**	54.36 ± 0.28
CV-MIM	✓	**58.61 ± 0.31**	**59.67 ± 0.29**	**52.53 ± 0.32**	**58.34 ± 0.25**	**57.79 ± 0.31**	52.08 ± 0.37	**56.50 ± 0.13**

Table 4. Classification accuracy (%) and standard deviation of models on NTU-RGB+D [19] with the setting of single-shot cross-view action recognition. C1, C2, and C3 are the camera identifiers; R1 and R2 are the replication numbers; one combination of them forms a unique camera view. Each time, models are trained using one view, and evaluated on all six views. We highlight view-disentangled (VD) representation learning methods. Best performances are highlighted in bold.

Methods	VD	2.0% (48)	3.0% (72)	5.0% (118)	7.5% (186)	10.0% (234)	12.5% (302)	15.0% (354)	20.0% (472)	All (2.31K)
Temporal ConvNet		58.7	66.7	83.1	87.1	90.5	91.4	91.7	93.5	**98.5**
Auto-Encoder	✓	59.8	72.2	83.6	88.2	90.6	91.8	92.9	94.1	97.7
VAE [11]	✓	60.6	71.7	81.9	87.4	91.1	91.7	92.9	94.0	97.6
β-VAE [1, 6]	✓	61.2	71.0	79.2	88.3	89.9	90.1	92.4	94.4	97.7
InfoNCE [15]	✓	59.8	69.8	81.1	85.2	90.6	90.9	92.3	93.9	97.5
DIM [7]	✓	57.6	68.1	80.0	84.3	87.7	90.0	90.6	94.3	97.3
CV-MIM	✓	**64.9**	**77.5**	**88.7**	**90.2**	**92.3**	**92.6**	**93.3**	**94.5**	98.1

Table 5. Comparisons of classification accuracy (%) on Penn Action [23] with the fully-supervised setting when the amount of training samples is varied. We highlight view-disentangled (VD) representation learning methods. Best performances are highlighted in bold.

[16] Xi Peng, Xiang Yu, Kihyuk Sohn, Dimitris N Metaxas, and Manmohan Chandraker. Reconstruction-based disentanglement for pose-invariant face recognition. In *Int. Conf. Comput. Vis.*, pages 1623–1632, 2017. 2, 3

[17] Helge Rhodin, Victor Constantin, Isinsu Katircioglu, Mathieu Salzmann, and Pascal Fua. Neural scene decomposition for multi-person motion capture. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7703–7713, 2019. 2, 3

[18] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3D human pose estimation. In *Eur. Conf. Comput. Vis.*, pages 750–767, 2018. 2, 3

[19] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3d human activity analysis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1010–1019, 2016. 3, 4, 5, 6

[20] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017. 2

[21] Jennifer J Sun, Jiaping Zhao, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, and Ting Liu. View-invariant probabilistic embedding for human pose. In *Eur. Conf. Comput. Vis.*, 2020. 5

[22] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, pages 7444–7452, 2018. 3

[23] Weiyu Zhang, Menglong Zhu, and Konstantinos G Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *Int. Conf. Comput. Vis.*, pages 2248–2255, 2013. 3, 4, 5, 6

Methods	VD	.5% (145)	1.0% (318)	2.0% (640)	3.0% (926)	5.0% (1.51K)	7.5% (2.31K)	10.0% (3.4K)	12.5% (3.86K)	All (30.7K)
Res-TCN [9]		20.7	31.2	40.6	46.8	55.1	62.4	64.7	71.2	**90.2**
Auto-Encoder	✓	22.4	27.5	40.5	45.6	55.4	59.2	64.5	67.1	71.8
VAE [11]	✓	22.9	31.6	42.3	49.4	55.0	61.9	64.4	69.4	88.7
β-VAE [1, 6]	✓	21.2	30.2	42.2	50.7	55.4	61.8	64.7	69.5	88.8
InfoNCE [15]	✓	20.8	26.5	37.8	43.0	49.8	55.1	58.3	62.1	82.7
DIM [7]	✓	20.2	23.8	35.0	40.0	47.6	52.8	57.0	59.9	82.4
CV-MIM	✓	**28.5**	**37.6**	**46.9**	**51.0**	**57.6**	**63.9**	**65.3**	**72.1**	89.5

Table 6. Results of top-1 classification accuracy (%) on NTU-RGB+D [19] with the cross-view benchmark when the amount of training samples is varied. We highlight view-disentangled (VD) representation learning methods. Best performances are highlighted in bold.

Methods	2D Pose	Auto-Encoder	VAE [11]	β-VAE [1, 6]	InfoNCE [15]	DIM [7]	**CV-MIM**
Accuracy (%)	62.13	66.24	66.49	65.63	66.14	65.71	**67.28**

Table 7. Results of view classification on Penn Action [23]. Best performances are highlighted in bold.

Figure 1. Nearest neighbors in the representation space using subjects S9 and S11 on Human3.6M [8]. The first two rows use pose representations, while the second two rows use view representations. On each row, we show the query on the left and its top five nearest neighbors on the right.

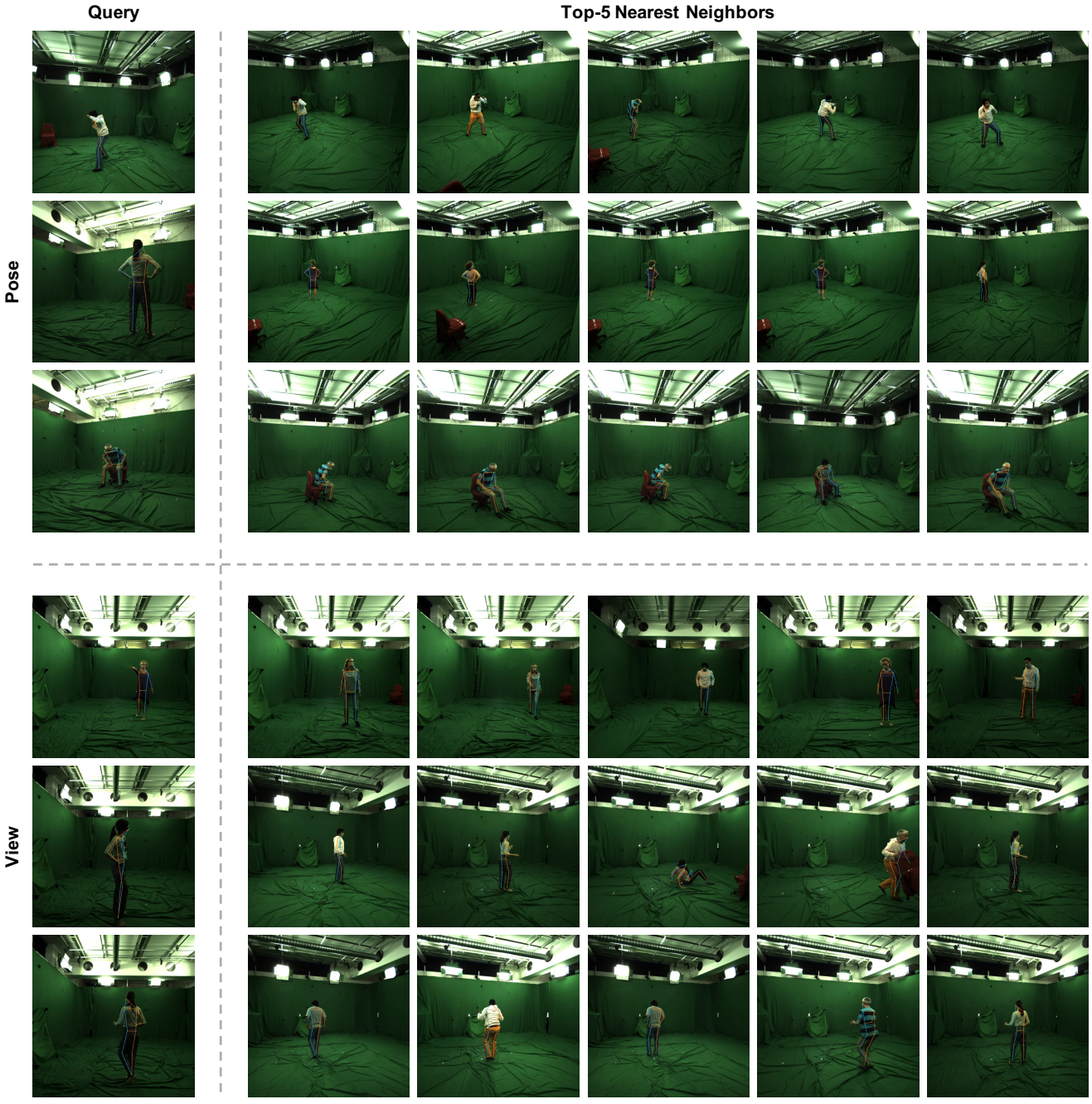

Figure 2. Nearest neighbors in the representation space on MPI-INF-3DHP [13]. The first three rows use pose representations, while the second three rows use view representations. On each row, we show the query on the left and its top five nearest neighbors on the right.