SA-DVAE: Improving Zero-Shot Skeleton-Based Action Recognition by Disentangled Variational Autoencoders

Sheng-Wei Li[1], Zi-Xiang Wei[2], Wei-Jie Chen[2], Yi-Hsin Yu[2], Chih-Yuan Yang[3,4], and Jane Yung-jen Hsu[2,3]

[1] Graduate Institute of Networking and Multimedia, National Taiwan University
[2] Department of Computer Science and Information Engineering, National Taiwan University
[3] Department of Artificial Intelligence, Chang Gung University
[4] Artificial Intelligence Research Center, Chang Gung University
{r11944004,r12922147,r12922051,r12922220,yjhsu}@csie.ntu.edu.tw, cyyang@cgu.edu.tw

Abstract. Existing zero-shot skeleton-based action recognition methods utilize projection networks to learn a shared latent space of skeleton features and semantic embeddings. The inherent imbalance in action recognition datasets, characterized by variable skeleton sequences yet constant class labels, presents significant challenges for alignment. To address the imbalance, we propose SA-DVAE—Semantic Alignment via Disentangled Variational Autoencoders, a method that first adopts feature disentanglement to separate skeleton features into two independent parts—one is semantic-related and another is irrelevant—to better align skeleton and semantic features. We implement this idea via a pair of modality-specific variational autoencoders coupled with a total correction penalty. We conduct experiments on three benchmark datasets: NTU RGB+D, NTU RGB+D 120 and PKU-MMD, and our experimental results show that SA-DAVE produces improved performance over existing methods. The code is available at `https://github.com/pha123661/SA-DVAE`.

Keywords: Skeleton-based Action Recognition · Zero-Shot and Generalized Zero-Shot Learning · Feature Disentanglement

1 Introduction

Action recognition is a long-standing active research area because it is challenging and has a wide range of applications like surveillance, monitoring, and human-computer interfaces. Based on input data types, there are several lines of studies on human action recognition: image-based, video-based, depth-based, and skeleton-based. In this paper, we focus on the skeleton-based action recognition, which is enabled by the advance in pose estimation [23,26] and sensor [13,27] technologies, and has emerged as a viable alternative to video-based action recognition due to its resilience to variations in appearance and background. Some

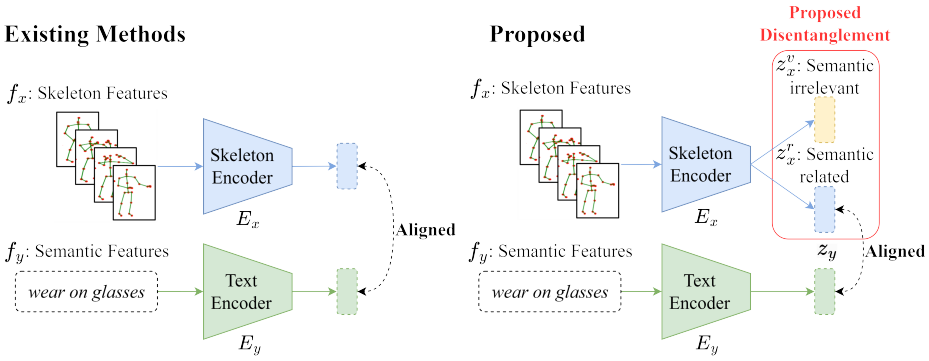


Fig. 1: Comparison with existing methods. Our method is the first to apply feature disentanglement to the problem of skeleton-based zero-shot action recognition. All existing methods directly align skeleton features with textual ones, but ours only aligns a part of semantic-related skeleton features with the textual ones.

existing skeleton-based action recognition methods already achieve remarkable performance on large-scale action recognition datasets [5, 16, 22] through supervised learning, but labeling data is expensive and time-consuming. For the cases where training data are difficult to obtain or prevented by privacy issues, zero-shot learning (ZSL) offers an alternative solution by recognizing unseen actions through supporting information such as the names, attributes, or descriptions of the unseen classes. Therefore, zero-shot learning has multiple types of input data and aims to learn an effective way of dealing with those data representations. For skeleton-based zero-shot action recognition, several methods have been proposed to align skeleton features and text features in the same space.

However, to the best of our knowledge, all existing methods assume that the group of skeleton sequences are well captured and highly consistent so their ideas mainly focus on how to semantically optimize text representation. After carefully examining the source videos in two widely used benchmark datasets NTU RGB+D and PKU-MMD, we found the assumption is questionable. We observe that for some labels, the camera positions and actors' action differences do bring in significant noise. To address this observation, we seek an effective way to deal with the problem. Inspired by an existing ZSL method [3] which shows semantic-irrelevant features can be separated from semantic-related ones, we propose SA-DVAE for skeleton-based action recognition. SA-DVAE tackles the generalization problem by disentangling the skeleton latent feature space into two components: a semantic-related term and a semantic-irrelevant term as shown in Fig. 1. This enables the model to learn more robust and generalizable visual embeddings by focusing solely on the semantic-related term for action recognition. In addition, SA-DVAE implements a learned total correlation penalty that encourages independence between the two factorized latent features and minimizes the shared information captured by the two representations. This

penalty is realized by an adversarial discriminator that aims to estimate the lower bound of the total correlation between the factorized latent features.

The contributions of our paper are as follows:

- We propose a novel SA-DVAE method. By disentangling the latent feature space into semantic-related and irrelevant terms, the model addresses the asymmetry existing in action recognition datasets and improves the generalization capability.
- We leverage an adversarial total correlation penalty to encourage independence between the two factorized latent features.
- We conduct extensive experiments that show SA-DVAE achieves state-of-the-art performance on the ZSL and generalized zero-shot learning (GZSL) benchmarks of the NTU RGB+D 60, NTU RGB+D 120, and PKU-MMD datasets.

2 Related Work

The proposed SA-DAVE method covers two research fields: zero-shot learning and action recognition, and it uses feature disentanglement to deal with skeleton data noise. Here we discuss the most related research reports in the literature.

Skeleton-Based Zero-Shot Action Recognition. ZSL aims to train a model under the condition that some classes are unseen during training. The more challenging GZSL expands the task to classify both seen and unseen classes during testing [18]. ZSL relies on semantic information to bridge the gap between seen and unseen classes.

Existing methods address the skeleton and text zero-shot action recognition problem by constructing a shared space for both modalities. ReViSE [12] learns autoencoders for each modality and aligns them by minimizing the maximum mean discrepancy loss between the latent spaces. Building on the concept of feature generation, CADA-VAE [21] employs variational autoencoders (VAEs) for each modality, aligning the latent spaces through cross-modal reconstruction and minimizing the Wasserstein distance between the inference models. These methods then learn classifiers on the shared space to conduct classification.

SynSE [7] and JPoSE [24] are two methods that leverage part-of-speech (PoS) information to improve the alignment between text descriptions and their corresponding visual representations. SynSE extends CADA-VAE by decomposing text descriptions by PoS tags, creating individual VAEs for each PoS label, and aligning them in the skeleton space. Similarly, JPoSE [24] learns multiple shared latent spaces for each PoS label using projection networks. JPoSE employs uni-modal triplet loss to maintain the neighborhood structure of each modality within the shared space and cross-modal triplet loss to align the two modalities.

On the other hand, SMIE [28] focuses on maximizing mutual information between skeleton and text feature spaces, utilizing a Jensen-Shannon Divergence

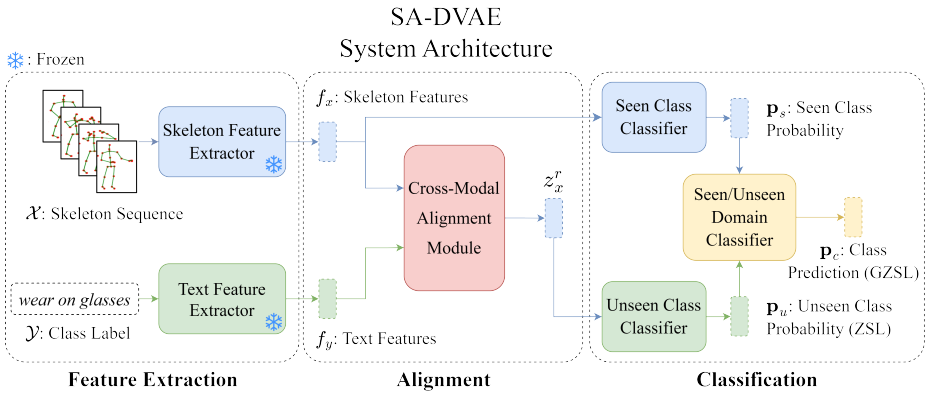


Fig. 2: System Architecture of SA-DVAE. Initially, the feature extractors are employed to extract features. Subsequently, the cross-modal alignment module aligns the two modalities and generates semantic-related unseen skeleton features (z_x^r). These generated features are utilized to train classifiers.

estimator trained with contrastive learning. It also considers temporal information in action sequences by promoting an increase in mutual information as more frames are observed.

While JPoSE and SynSE demonstrate the benefits of incorporating PoS information, they rely heavily on it and require additional PoS tagging effort. Furthermore, the two methods neglect the inherent asymmetry between modalities, aligning semantic-related and irrelevant terms to the semantic features and missing the chance to improve recognition accuracy further. In contrast, our approach uses simple class labels without the need of PoS tags, and uses only semantic-related skeleton information to align text data.

Feature Disentanglement in Generalized Zero-Shot Learning. Feature disentanglement refers to the process of separating the underlying factors of variation in data [2]. Because methods of zero-shot learning are sensitive to the quality of both visual and semantic features, feature disentanglement serves as an effective approach to scrutinize either visual or semantic features, as well as addressing the domain shift problem [18], thereby generating more robust and generalized representations.

SDGZSL [3] decomposes visual embeddings into semantic-consistent and semantic-unrelated components using shared class-level attributes, and learns an additional relation network to maximize compatibility between semantic-consistent representations and their corresponding semantic embeddings. This approach is motivated by the transfer of knowledge from intermediate semantics (e.g., class attributes) to unseen classes. In contrast, SA-DVAE addresses the inherent asymmetry between the text and skeleton modalities, enabling the direct use of text descriptions instead of relying on predefined class attributes.

3 Methodology

We show the overall architecture of our method as Fig. 2, which consists of three main components: a) two modality-specific feature extractors, b) a cross-modal alignment module, and c) three classifiers for seen/unseen actions and their domains. The cross-modal alignment module learns a shared latent space via cross-modality reconstruction, where feature disentanglement is applied to prioritize the alignment of semantic-related information (z_x^r and z_y). To improve the effectiveness of the disentanglement, we use a discriminator as an adversarial total correlation penalty between the disentangled features.

Problem Definition. Let $\mathcal{D}$ be a skeleton-based action dataset consisting of a skeleton sequences set $\mathcal{X}$ and a label set $\mathcal{Y}$, in which a label is a piece of text description. The $\mathcal{X}$ is split into a seen and unseen subset $\mathcal{X}_s$ and $\mathcal{X}_u$ where we can only use $\mathcal{X}_s$ and $\mathcal{Y}$ to train a model to classify $x \in \mathcal{X}_u$. By definition, there are two types of evaluation protocols. The GZSL one asks to predict the class of x among all classes $\mathcal{Y}$, and the ZSL only among $\mathcal{Y}_u = \{y_i : x_i \in \mathcal{X}_u\}$.

Cross-Modal Alignment Module. We train a skeleton representation model (Shift-GCN [4] or ST-GCN [25], depending on experimental settings) on the seen classes using standard cross-entropy loss. This model extracts our skeleton features, denoted as f_x. We use a pre-trained language model (Sentence-BERT [20] or CLIP [19]) to extract our label's text features, denoted as f_y. Because f_x and f_y belong to two unrelated modalities, we train two modality-specific VAEs to adjust f_x and f_y for our recognition task and illustrate their data flow in Fig. 3. Our encoders E_x and E_y transform f_x and f_y into representations z_x and z_y in a shared latent space via the reparameterization trick [14]. To optimize the VAEs, we introduce a loss as the form of the Evidence Lower Bound

$$\mathcal{L} = \mathbb{E}_{q_\phi(z|f)}[\log p_\theta(f|z)] - \beta D_{KL}(q_\phi(z|f) \| p_\theta(z)), \tag{1}$$

where β is a hyperparameter, f and z are the observed data and latent variables, the first term is the reconstruction error, and the second term is the Kullback-Leibler divergence between the approximate posterior $q(z|f)$ and $p(z)$. The hyperparameter β balances the quality of reconstruction with the alignment of the latent variables to a prior distribution [8]. We use multivariate Gaussian as the prior distribution.

Feature Disentanglement. We observe that although two skeleton sequences belong to the same class (*i.e.* they share the same text description), their movement varies substantially due to stylistic factors such as actors' body shapes and movement ranges, and cameras' positions and view angles. To the best of our knowledge, existing methods never address this issue. For example, Zhou *et al.* [28] and Gupta *et al.* [7] neglect this issue and force f_x and f_y to be aligned. Therefore, we propose to tackle the problem of inherent asymmetry between the two modalities to improve the recognition performance.

We design our skeleton encoder E_x as a two-head network, of which one head generates a semantic-related latent vector z_x^r and the other generates a semantic-irrelevant vector z_x^v. We assume each of z_x^r and z_x^v has its own multivariant normal

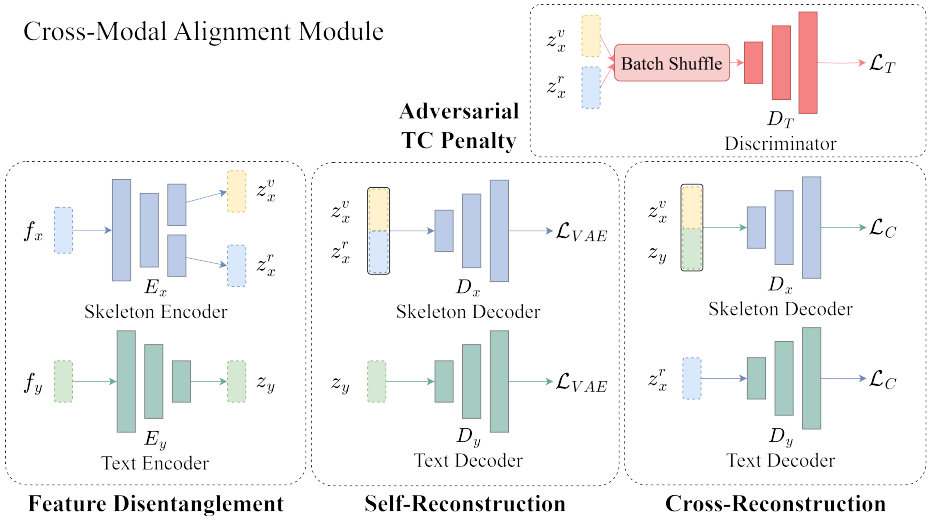


Fig. 3: Cross-Modal Alignment Module. This module serves two primary tasks: latent space construction through self-reconstruction and cross-modal alignment via cross-reconstruction. The skeleton features are disentangled into semantic-related (z_x^r) and irrelevant (z_x^v) factors.

distribution $N(\mu_x^r, \Sigma_x^r)$ and $N(\mu_x^v, \Sigma_x^v)$, and our text encoder E_y generates a latent feature z_y, which also has a multivariant normal distribution $N(\mu_y, \Sigma_y)$.

Let $z_x = z_x^v \oplus z_x^r$ where $\oplus$ means concatenation. We define the losses for the VAEs as

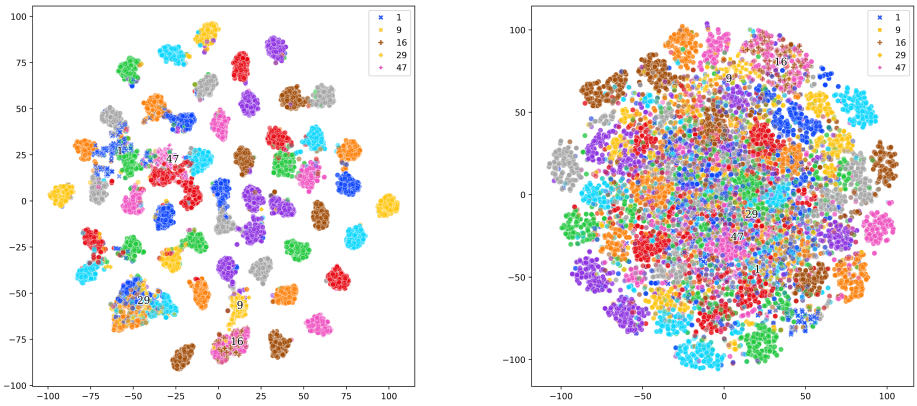
$$\begin{aligned}\mathcal{L}_x = &\mathbb{E}_{q_\phi(z_x|f_x)}[\log p_\theta(f_x|z_x)] \\ &- \beta_x D_{KL}(q_\phi(z_x^r|f_x)||p_\theta(z_x^r)) \\ &- \beta_x D_{KL}(q_\phi(z_x^v|f_x)||p_\theta(z_x^v)),\end{aligned} \tag{2}$$

$$\mathcal{L}_y = \mathbb{E}_{q_\phi(z_y|f_y)}[\log p_\theta(f_y|z_y)] - \beta_y D_{KL}(q_\phi(z_y|f_y)||p_\theta(z_y)), \tag{3}$$

where β_x and β_y are hyperparameters, $p_\theta(z_x^r)$, $p_\theta(z_x^v)$, $p_\theta(f_x|z_x)$, $p_\theta(z_y)$, and $p_\theta(f_y|z_y)$ are the probabilities of their presumed distributions, $q_\phi(z_x|f_x)$, $q_\phi(z_x^r|f_x)$ and $q_\phi(z_x^v|f_x)$ are the probabilities calculated through our skeleton encoder E_x, and $q_\phi(z_y|f_y)$ is the one through our text encoder E_y. We set the overall VAE loss as

$$\mathcal{L}_{VAE} = \mathcal{L}_x + \mathcal{L}_y. \tag{4}$$

To better understand our method, we present the t-SNE visualization of the semantic-related and semantic-irrelevant terms, z_x^r and z_x^v in Fig. 4. Figure 4a displays the t-SNE results for z_x^r, showing clear class clusters that demonstrate effective disentanglement. In contrast, Figure 4b shows the t-SNE results for z_x^v, where class separation is less distinct. This indicates that while our method effectively clusters related semantic features, the irrelevant features remain more dispersed as they contain instance-specific information.



(a) t-SNE visualization of z_x^r. **(b)** t-SNE visualization of z_x^v.

Fig. 4: t-SNE visualizations of z_x^r and z_x^v. Best viewed in color.

Cross-Alignment Loss. Because we want our latent text features z_y to align with semantic-related skeleton features z_x^r only, regardless of the semantic-irrelevant features z_x^v, we regulate them by setting up a cross-alignment loss

$$\mathcal{L}_C = \|D_y(z_x^r) - f_y\|_2^2 + \|D_x(z_x^v \oplus z_y) - f_x\|_2^2 \tag{5}$$

to train our VAEs for skeleton and text respectively. This loss enforces skeleton features to be reconstructable from text features and vice versa. To reconstruct skeleton features from text features, z_x^v is employed to incorporate necessary style information to mitigate the information gap between the class label and the skeleton sequence.

Adversarial Total Correlation Penalty. We expect the features z_x^r and z_x^v to be statistically independent, so we impose an adversarial total correlation penalty [3] on them. We train a discriminator D_T to predict the probability of a given latent skeleton vector $z_x^v \oplus z_x^r$ whether the z_x^v and z_x^r come from the same skeleton feature f_x. In the ideal case, D_T will return 1 if z_x^v and z_x^r are generated together, and 0 otherwise. To train D_T, we design a loss

$$\mathcal{L}_T = \log D_T(z_x) + \log(1 - D_T(\tilde{z}_x)), \tag{6}$$

where $\tilde{z}_x$ is an altered feature vector. We create $\tilde{z}_x$ as the following steps. From a batch of N training samples, our encoder E_x generates N pairs of $z_{x,i}^v$ and $z_{x,i}^r$, $i = 1 \dots N$. We randomly permute the indices i of $z_{x,i}^v$ but keep $z_{x,i}^r$ unchanged, and then we concatenate them as $\tilde{z}_x$. D_T is trained to maximize L_T, while E_x is adversarially trained to minimize it. This training process encourages the encoder to generate latent representations that are independent. Combining the three losses, we set the overall loss

$$\mathcal{L} = \mathcal{L}_{VAE} + \lambda_1 \mathcal{L}_C + \lambda_2 \mathcal{L}_T, \tag{7}$$

where we balance the three losses by hyperparameters λ_1 and λ_2.

Seen, Unseen and Domain Classifier. Because there are two protocols, ZSL and GZSL, to evaluate a zero-shot recognition model, we use two different settings for the two protocols. For the ZSL protocol, we only need to predict the probabilities of classes $\mathcal{Y}_u$ from a given skeleton sequence, so we propose a classifier C_u as a single-layer MLP (Multilayer Perception) with a softmax output layer yielding the probabilities to predict probabilities of classes $\mathcal{Y}_u$ from z_y by

$$\mathbf{p}_u = C_u(z_y) = C_u(E_y(f_y)), \tag{8}$$

where $\dim(\mathbf{p}_u) = |\mathcal{Y}_u|$. During inference and given an unseen skeleton feature f_x^u, we get $z_x^u = E_x(f_x^u)$, separate z_x^u into $z_x^{v,u}$ and $z_x^{r,u}$, and generate $\mathbf{p}_u = C_u(z_x^{r,u})$ to predict its class as $y_{\hat{i}}$ and

$$\hat{i} = \underset{i=1,\ldots,|\mathcal{Y}_u|}{\arg\max}\ p_u^i, \tag{9}$$

where p_u^i is the i-th probability value of $\mathbf{p}_u$.

For the GZSL protocol, we need to predict the probabilities of all classes in $\mathcal{Y} = \mathcal{Y}_u \cup \mathcal{Y}_s$ where $\mathcal{Y}_s = \{y_i : x_i \in \mathcal{X}_s\}$. We follow the same approach proposed by Gupta *et al.* [7] to use an additional class classifier C_s for seen classes and a domain classifier C_d to merge two arrays of probabilities. Gupta *et al.* first apply Atzmon and Chechik's idea [1] to a skeleton-based action recognition problem and outperform the typical single-classifier approach. The advantage of using dual classifiers is reported in a review paper [18]. Our C_s is also a single-layer MLP with a softmax output layer like C_u, but it uses skeleton features f_x rather than latent features to produce probabilities

$$\mathbf{p}_s = C_s(f_x), \tag{10}$$

where $\dim(\mathbf{p}_s) = |\mathcal{Y}_s|$.

We train C_s and C_u first, and then we freeze their parameters to train C_d, which is a logistic regression with an input vector $\mathbf{p}'_s \oplus \mathbf{p}_u$ where $\mathbf{p}'_s$ is the temperature-tuned [9] top k-pooling result of $\mathbf{p}_s$ and the number $k = \dim(\mathbf{p}_u)$. C_d yields a probability value p_d of whether the source skeleton belongs to a seen class. We use the LBFGS algorithm [15] to train C_d and use it during inference to predict the probability of x as

$$\mathbf{p}(y|x) = C_d(\mathbf{p}'_s \oplus \mathbf{p}_u)\mathbf{p}_s \oplus (1 - C_d(\mathbf{p}'_s \oplus \mathbf{p}_u))\mathbf{p}_u = p_d\mathbf{p}_s \oplus (1 - p_d)\mathbf{p}_u \tag{11}$$

and decide the class of x as $y_{\hat{i}}$ and

$$\hat{i} = \underset{i=1,\ldots,|\mathcal{Y}|}{\arg\max}\ p^i, \tag{12}$$

where p^i is the i-th probability value of $\mathbf{p}(y|x)$.

4 Experiments

Datasets. We conduct experiments on three datasets and show their statistics in Table 1. We adopt the cross-subject split, where half of the subjects are

Table 1: Statistics of datasets used in our experiments

Name	Class	Subject	Joint	Sample	Camera View
NTU RGB+D [22]	60	40	25	56,880	3
NTU RGB+D 120 [16]	120	106	25	114,480	3
PKU-MMD [5]	51	66	25	28,443	3

used for training and the other half for validation. We use NTU-60 and NTU-120 as synonyms for the NTU RGB+D and NTU RGB+D 120 datasets. Due to discrepancies in class labels between the official website[5] and the GitHub codebase[6] of NTU-60 and NTU-120 datasets (*e.g.* the label of class 18 is "put on glasses" in their website but "wear on glasses" in GitHub), we follow existing methods by using the class labels provided in their codebase.

Implementation Details. We implement the discriminator D_T as a two-layer MLP with ReLU activation and a Sigmoid output layer, and the encoders E_x, E_y, decoders D_x, D_y, seen and unseen classifiers C_s, C_u as single-layer MLPs. During training, we alternatively train VAEs and D_T. We train VAEs first, and after training VAEs n_d times, we train D_T once.

We use the LBFGS implementation from Scikit-learn [17] to train C_d and divide our training set into a validation seen set and a validation unseen set. As the training of C_d requires seen and unseen skeleton features (f_x^s, f_x^u), we re-train other components using the validation seen set and use the validation unseen set to provide unseen skeleton features to train C_d. Finally, the trained C_d is used to make inferences on the testing set. The number of classes in the validation unseen set is the same as the original unseen class set $|\mathcal{Y}_u|$.

We use the cyclical annealing schedule [6] to train our VAEs because cyclical annealing mitigates the KL divergence vanishing problem. At the beginning of each epoch, we set the actual training hyperparameters λ_2', β_1', and β_2' as 0 until we use one-third training samples. Thereafter, we progressively increase λ_2', β_1', and β_2' to λ_2, β_x, and β_y based on the number of trained samples, e.g,

$$\lambda_2' = \begin{cases} 0 & \text{if } k < \frac{1}{3}n; \\ \frac{3}{2}(\frac{k}{n} - \frac{1}{3})\lambda_2 & \text{if } k \geq \frac{1}{3}n, \end{cases} \tag{13}$$

where k and n are the index and total number of training samples in an epoch. We set λ_1 as 0 in our first epoch and 1 for all subsequent epochs. We conduct our experiments on a machine equipped with an Intel i7-13700 CPU, an NVIDIA RTX 3090 GPU, and 32GB RAM. We implement our method using PyTorch 2.1.0, scikit-learn 1.3.2, and scipy 1.11.3. It takes 4.6 hours to train our model for a 55/5 split of the NTU RGB+D 60 dataset, and 8.7 hours for a 110/10 split of the NTU RGB+D 120 dataset. We determine the hyperparameters through random search, as listed in Tables 2 and 5. The hyperparameter search space is detailed in Supplementary Materials Section A.

[5] Official website: `https://rose1.ntu.edu.sg/dataset/actionRecognition/`

[6] GitHub link: `https://github.com/shahroudy/NTURGB-D`

Table 2: Setting for comparison with existing methods.

	NTU-60	NTU-120
Skeleton Feature Extractor	Shift-GCN [4]	
Text Feature Extractor	Sentence-BERT [20]	
Epochs	10	
Optimizer	Adam	
Optimizer Momentum	$\beta_1 = 0.9, \beta_2 = 0.999$	
Batch size	32	
Learning rate	3.39e-05	3.48e-05
Weights of D_{KL} in $\mathcal{L}_{VAE}$	$\beta_x = 0.023, \beta_y = 0.011$	
Weight of $\mathcal{L}_T$	$\lambda_2 = 0.011$	
Discriminator steps n_d	5	4
Hidden dim. of z_x^r and z_y	160	256
Hidden dim. of z_x^v	8	32

Table 3: ZSL accuracy (%) on the NTU RGB+D datasets.

Method	NTU-60		NTU-120	
	55/5 split	48/12 split	110/10 split	96/24 split
ReViSE [12]	53.91	17.49	55.04	32.38
JPoSE [24]	64.82	28.75	51.93	32.44
CADA-VAE [21]	76.84	28.96	59.53	35.77
SynSE [7]	75.81	33.30	62.69	38.70
SMIE [28]	77.98	40.18	65.74	45.30
SA-DVAE	**82.37**	**41.38**	**68.77**	**46.12**

Comparison with SOTA methods. We compare our method with several state-of-the-art zero-shot action recognition methods using the setting shown in Table 2 and report their results in Tables 3 and 4. We use the same feature extractors and class splits as the one used by SynSE, and the only difference lies in the network architecture.

The results show that SA-DVAE works well, in particular for unseen classes. Furthermore, for the more challenging GZSL task, SA-DVAE even improves more over existing methods. On the NTU RGB+D 60 dataset, SA-DVAE improves the accuracy of (+7.25% and +6.23%) in the GZSL protocol, greater than the (+4.39% and +1.2%) in the ZSL one.

Random Class Splits and Improved Feature Extractors. The setting of class splits is crucial for accuracy calculation and Tables 3 and 4 only show results of a few predefined splits, which can not infer the overall performance on a complete dataset. Thus, we follow Zhou *et al.*'s approach [28] to randomly select several unseen classes as a new split, repeat it three times, and report the average performance. In addition, we use improved skeleton feature extractor ST-GCN [25] and text extractor CLIP [19], chosen for their broad applicability

Table 4: GZSL metrics: seen class accuracy Acc_s, unseen class accuracy Acc_u, and their harmonic mean H (%) on the NTU RGB+D datasets. *: SynSE paper reports 29.22, but it is a miscalculation.

Method	NTU-60						NTU-120					
	55/5 split			48/12 split			110/10 split			96/24 split		
	Acc_s	Acc_u	H	Acc_s	Acc_u	H	Acc_s	Acc_u	H	Acc_s	Acc_u	H
ReViSE [12]	74.22	34.73	47.32*	62.36	20.77	31.16	48.69	44.84	46.68	49.66	25.06	33.31
JPoSE [24]	64.44	50.29	56.49	60.49	20.62	30.75	47.66	46.40	47.05	38.62	22.79	28.67
CADA-VAE [21]	69.38	61.79	65.37	51.32	27.03	35.41	47.16	49.78	48.44	41.11	34.14	37.31
SynSE [7]	61.27	56.93	59.02	52.21	27.85	36.33	52.51	57.60	54.94	56.39	32.25	41.04
SA-DVAE	62.28	70.80	**66.27**	50.20	36.94	**42.56**	61.10	59.75	**60.42**	58.82	35.79	**44.50**

Table 5: Settings for the random-split experiment.

	NTU-60	NTU-120	PKU-MMD
Skeleton Feature Extractor		ST-GCN [25]	
Text Feature Extractor		CLIP-ViT-B/32 [19]	
Epochs		10	
Optimizer		Adam	
No. of unseen classes	5	10	5
Optimizer Momentum		$\beta_1 = 0.9, \beta_2 = 0.999$	
Batch size	32	32	64
Learning rate	3.60e-05	7.36e-5	3.07e-05
Weights of D_{KL} in $\mathcal{L}_{VAE}$		$\beta_x = 0.023, \beta_y = 0.011$	
Weight of $\mathcal{L}_T$		$\lambda_2 = 0.011$	
Discriminator steps n_d	7	12	2
Hidden dim. of z_x^r and z_y	224	176	128
Hidden dim. of z_x^v	16	20	16

and robust performance across different domains. We also tested different feature extractors, which can be found in Supplementary Materials Section B.

Table 5 shows our settings and Tables 6 and 7 show the results, where naive alignment means that we disable D_T and remove the extra head for z_x^v, and FD means that we disable D_T. The results show that both feature disentanglement and total correlation penalty contribute to accuracy improvements, and feature disentanglement is the major contributor, *e.g.*, +12.95% on NTU-60 compared to naive alignment in Table 6. The adversarial total correlation penalty (TC) slightly reduces the accuracy for seen classes but significantly improves unseen and overall accuracy. This is because TC enhances the embedding quality by reducing feature redundancy, making the domain classifier less biased towards seen classes. Consequently leading to improved generalization. The results in Tab. 7 highlight this trade-off, where the improved harmonic mean indicates a more balanced and robust performance across both seen and unseen classes.

Table 6: Average ZSL accuracy (%) under the random split setting on the NTU-60, NTU-120, and PKU-MMD datasets. FD: feature disentanglement. TC: adversarial total correlation penalty. †: PoS tags for the PKU-MMD dataset are obtained from spaCy [11].

Method	NTU-60 55/5 split	NTU-120 110/10 split	PKU-MMD 46/5 split
ReViSE [12]	60.94	44.90	59.34
JPoSE† [24]	59.44	46.69	57.17
CADA-VAE [21]	61.84	45.15	60.74
SynSE† [7]	64.19	47.28	53.85
SMIE [28]	65.08	46.40	60.83
Naive alignment	69.26	39.73	60.13
FD	82.21	49.18	60.97
SA-DVAE (FD+TC)	**84.20**	**50.67**	**66.54**

Table 7: Average GZSL metrics: seen class accuracy Acc_s, unseen class accuracy Acc_u, and their harmonic mean H (%) under the random split setting on the NTU-60, NTU-120, and PKU-MMD datasets. FD: feature disentanglement. TC: adversarial total correlation penalty. †: PoS tags for the PKU-MMD dataset are obtained from spaCy [11].

Method	NTU-60 55/5 splits			NTU-120 110/10 split			PKU-MMD 46/5 split		
	Acc_s	Acc_u	H	Acc_s	Acc_u	H	Acc_s	Acc_u	H
ReViSE [12]	71.75	52.06	60.34	48.29	34.64	40.34	60.89	42.16	49.82
JPoSE † [24]	66.25	54.92	60.05	49.43	39.14	43.69	60.26	45.18	51.64
CADA-VAE [21]	77.35	58.14	66.38	51.09	41.24	45.64	63.17	35.86	45.75
SynSE † [7]	75.84	60.77	67.47	41.73	45.36	43.47	63.09	40.69	49.47
Naive alignment	82.11	47.99	60.58	57.01	31.62	40.68	58.76	43.14	49.75
FD	82.31	61.98	70.71	58.57	37.83	45.97	58.11	48.15	52.66
SA-DVAE (FD+TC)	78.16	72.60	**75.27**	58.09	40.23	**47.54**	58.49	51.40	**54.72**

From our three runs of the random-split experiment on the NTU-60 dataset (average results is shown in Table 6), we pick the most challenging run and show its per-class accuracy in Fig. 5 and the t-SNE visualization of skeleton features (f_x) in Fig. 6. The labels of classes 16 and 17 are "wear a shoe" and "take off a shoe" and their movements are acted as a person sitting on a chair who bends down her upper body and stretches her arm to touch her shoe. The skeleton sequences of the two classes are highly similar so are their extracted features. In Fig. 6, samples of classes 16 and 17 are overlapped, and naive alignment generates poor accuracy on class 16. Similarly, naive alignment generates near-zero accuracy on classes 9 and 29. Since both classes 9 and 16 share similar skeleton sequences and were unseen during training, their features appear highly similar. This similarity leads naive alignment to misclassify samples belonging

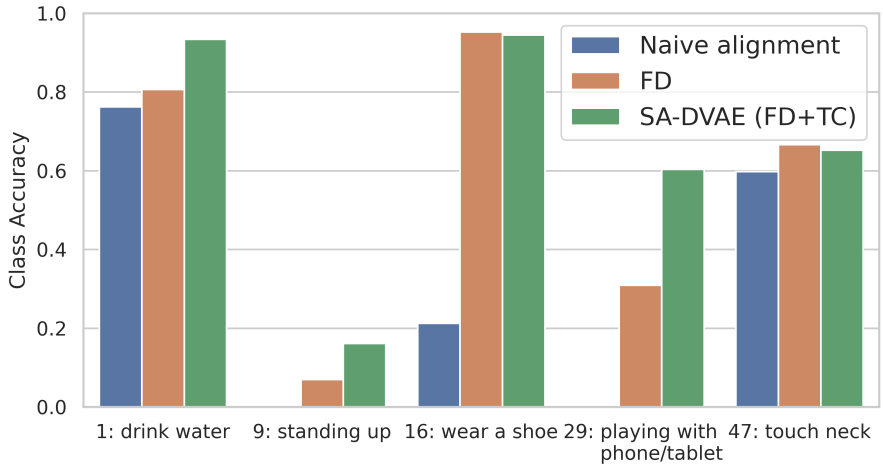


Fig. 5: Unseen per-class accuracy of the NTU-60 dataset. The unseen split {1, 9, 16, 29, 47} is used in a challenging run of our random-split GZSL experiments.

Table 8: Average GZSL metrics (%) of different seen classifier input under the random split setting on the NTU-60, NTU-120, and PKU-MMD datasets.

Method	NTU-60 55/5 splits			NTU-120 110/10 split			PKU-MMD 46/5 split		
	Acc_s	Acc_u	H	Acc_s	Acc_u	H	Acc_s	Acc_u	H
SA-DVAE (z_x^r as input)	72.00	71.48	71.74	55.35	39.00	45.76	59.16	49.73	54.04
SA-DVAE (f_x as input)	78.16	72.60	**75.27**	58.09	40.23	**47.54**	58.49	51.40	**54.72**

to class 9 as class 16. We can see significant improvements with the addition of FD and TC. These techniques allow the model to prioritize semantic-related information and improve classification performance.

Impact of Replacing Skeleton Feature f_x with Semantic-Related Latent Vector z_x^r in Seen Classifier We replace the input skeleton feature f_x of the seen classifier with the disentangled semantic-related latent vector z_x^r under the random-split setting listed in Tab. 5 and report results in Tab. 8. Notably, since the semantic-irrelevant terms also contain information that is beneficial for classification but not necessary related to the text descriptions, f_x retains both semantic-related and irrelevant details. This dual retention enhances performance compared to z_x^r, which focuses solely on semantic-related information.

We incorporate zero-shot learning and action recognition techniques, including pose canonicalization [10] and enhanced action descriptions [28], with additional experimental results in Supplementary Materials Section C.

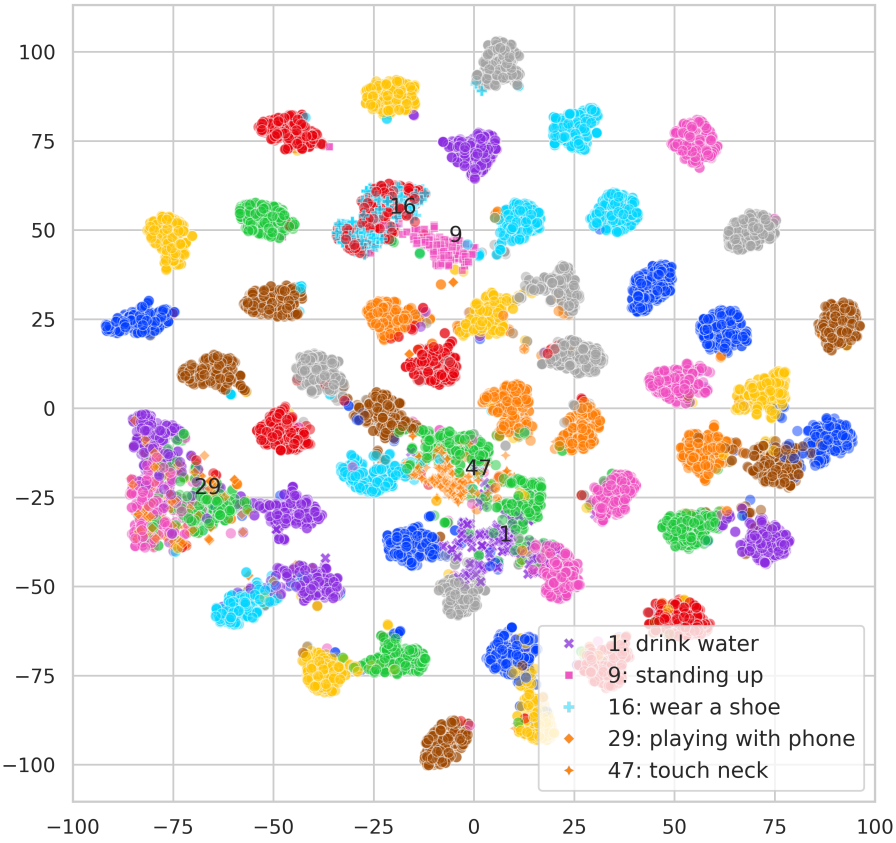


Fig. 6: t-SNE visualization of f_x of the NTU-60 dataset. The unseen split {1, 9, 16, 29, 47} is used in a run of our random-split GZSL experiments. Best viewed in color.

5 Conclusion

ZSL study aims to leverage knowledge from one domain to help solve problems in another domain and has been proven useful for action recognition tasks, in particular for 3D skeleton data because it is expensive and labor-consuming to build accurately labeled datasets. Although there are several existing methods in the literature, they never address the asymmetry problem between skeleton data and text description. In this paper, we propose SA-DVAE, a cross-modality alignment model using the feature disentanglement approach to differentiate skeleton data into two independent representations, the semantic-related and irrelevant ones. Along with an adversarial discriminator to enhance the feature disentanglement, our experiments show that the proposed method generates better performance over existing methods on three benchmark datasets in both ZSL and GZSL protocols.

Acknowledgments

This research was supported by the National Science and Technology Council of Taiwan under grant number 111-2622-8-002-028. The authors would like to thank the NSTC for its generous support.

References

1. Atzmon, Y., Chechik, G.: Adaptive confidence smoothing for generalized zero-shot learning. In: CVPR (2019)
2. Bengio, Y.: Deep learning of representations: Looking forward. In: Statistical Language and Speech Processing (2013)
3. Chen, Z., Luo, Y., Qiu, R., Wang, S., Huang, Z., Li, J., Zhang, Z.: Semantics disentangling for generalized zero-shot learning. In: ICCV (2021)
4. Cheng, K., Zhang, Y., He, X., Chen, W., Cheng, J., Lu, H.: Skeleton-based action recognition with shift graph convolutional network. In: CVPR (2020)
5. Chunhui, L., Yueyu, H., Yanghao, L., Sijie, S., Jiaying, L.: PKU-MMD: A large scale benchmark for continuous multi-modal human action understanding. arXiv preprint arXiv:1703.07475 (2017)
6. Fu, H., Li, C., Liu, X., Gao, J., Celikyilmaz, A., Carin, L.: Cyclical annealing schedule: A simple approach to mitigating KL vanishing. arXiv preprint arXiv:1903.10145 (2019)
7. Gupta, P., Sharma, D., Sarvadevabhatla, R.K.: Syntactically guided generative embeddings for zero-shot skeleton action recognition. In: ICIP (2021)
8. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. In: ICLR (2016)
9. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
10. Holden, D., Saito, J., Komura, T.: A deep learning framework for character motion synthesis and editing. ACM Transactions on Graphics (TOG) **35**(4), 1–11 (2016)
11. Honnibal, M., Montani, I., Van Landeghem, S., Boyd, A.: spaCy: Industrial-strength Natural Language Processing in Python (2020). `https://doi.org/10.5281/zenodo.1212303`
12. Hubert Tsai, Y.H., Huang, L.K., Salakhutdinov, R.: Learning robust visual-semantic embeddings. In: ICCV (2017)
13. Keselman, L., Iselin Woodfill, J., Grunnet-Jepsen, A., Bhowmik, A.: Intel RealSense stereoscopic depth cameras. In: CVPRW (2017)
14. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
15. Liu, D.C., Nocedal, J.: On the limited memory bfgs method for large scale optimization. Mathematical programming **45**(1-3), 503–528 (1989)
16. 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. TPAMI **42**(10), 2684–2701 (2019)
17. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)

18. Pourpanah, F., Abdar, M., Luo, Y., Zhou, X., Wang, R., Lim, C.P., Wang, X.Z., Wu, Q.J.: A review of generalized zero-shot learning methods. TPAMI **45**(4), 4051–4070 (2022)
19. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
20. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using Siamese BERT-networks. arXiv preprint arXiv:1908.10084 (2019)
21. Schonfeld, E., Ebrahimi, S., Sinha, S., Darrell, T., Akata, Z.: Generalized zero-and few-shot learning via aligned variational autoencoders. In: CVPR (2019)
22. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: NTU RGB+D: A large scale dataset for 3D human activity analysis. In: CVPR (2016)
23. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: CVPR (2019)
24. Wray, M., Larlus, D., Csurka, G., Damen, D.: Fine-grained action retrieval through multiple parts-of-speech embeddings. In: ICCV (2019)
25. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: AAAI. vol. 32 (2018)
26. Yuan, Y., Fu, R., Huang, L., Lin, W., Zhang, C., Chen, X., Wang, J.: HRFormer: High-resolution vision transformer for dense prediction. In: NeurIPS (2021)
27. Zhang, Z.: Microsoft Kinect sensor and its effect. IEEE Multimedia **19**(2), 4–10 (2012)
28. Zhou, Y., Qiang, W., Rao, A., Lin, N., Su, B., Wang, J.: Zero-shot skeleton-based action recognition via mutual information estimation and maximization. In: ACM MM (2023)