Alignment-Uniformity aware Representation Learning for Zero-shot Video Classification

Shi Pu[1]* Kaili Zhao[2]* Mao Zheng[1]
[1]Tencent AI Platform Department [2]Beijing University of Posts and Telecom.
{shipu, moonzheng}@tencent.com kailizhao@bupt.edu.cn

Abstract

*Most methods tackle zero-shot video classification by aligning visual-semantic representations within seen classes, which limits generalization to unseen classes. To enhance model generalizability, this paper presents an end-to-end framework that preserves alignment and uniformity properties for representations on both seen and unseen classes. Specifically, we formulate a supervised contrastive loss to simultaneously align visual-semantic features (*i.e.*, alignment) and encourage the learned features to distribute uniformly (*i.e.*, uniformity). Unlike existing methods that only consider the alignment, we propose uniformity to preserve maximal-info of existing features, which improves the probability that unobserved features fall around observed data. Further, we synthesize features of unseen classes by proposing a class generator that interpolates and extrapolates the features of seen classes. Besides, we introduce two metrics, closeness and dispersion, to quantify the two properties and serve as new measurements of model generalizability. Experiments show that our method significantly outperforms SoTA by relative improvements of 28.1% on UCF101 and 27.0% on HMDB51. Code is available*[1].

1. Introduction

Mimicking human capability to recognize things never seen before, zero-shot video classification (ZSVC) only trains models on videos of seen classes and makes predictions on unobserved ones [13, 19, 24, 27, 28, 51, 52, 54]. Correspondingly, existing ZSVC models map visual and semantic features into a unified representation, and hope the association can be generalized to unseen classes [2, 3, 6, 14, 35, 54]. However, these methods learn associated representations within limited classes, thus facing the following two critical problems [11, 13]: (1) semantic-gap: manifolds inconsistency between visual and semantics features, and (2) domain-shift: the representations learned from training sets are biased when applied to the target sets due to disjoint classes between two groups. In ZSVC, these two problems cause side effects on model generalizability.

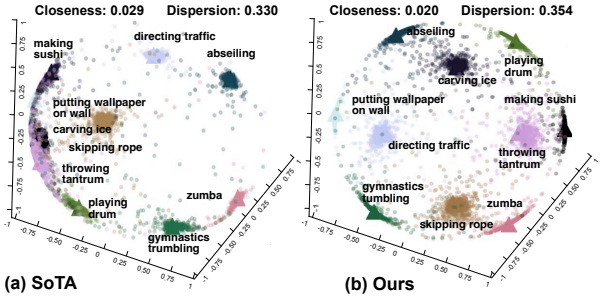


Figure 1. **Visual-semantic representations:** Comparisons of the learned representations between the SoTA [3] and our method. • and △ represent visual and semantic features separately; colors are for different classes. Besides, we use two metrics to quantify feature qualities on alignment (closeness↓ better) and uniformity (dispersion↑ better). We observe that ours show better closeness within classes and more separations among semantic clusters.

Reviewing the literature, we observe that most methods focus on tackling the semantic-gap by learning alignment-aware representations, which ensure visual and semantic features of the same class close. To improve the alignment, MSE loss [3], ranking loss [14], and center loss [13] are commonly used to optimize the similarity between visual and semantic features. Apart from the loss, improvements for alignment are attributed mainly to the designs of architectures. For instance, [13, 16, 28] first project global visual features to local object attributes, then optimize similarity between the attributes and final semantics. In contrast, URL [54], Action2Vec [14], and TARN [2] directly align visual and final semantic features, which are improved via attention modules. Since video features are hard to learn, the above methods utilize pre-trained models to extract visual features. The recent model [3] benefits from the efficient R(2+1)D module [43] in video classification and achieves the state-of-the-art (SoTA) results in ZSVC. However, the SoTA [3] neglects to learn semantic features; thus, it is still

*These authors contributed equally.
[1]https://github.com/ShipuLoveMili/CVPR2022-AURL

not a true end-to-end (e2e) framework for visual-semantic feature learning. We claim that e2e is critical for alignment since fixed visual/semantic features will bring obstacles to adjusting one to approach another.

Noteworthily, the latest MUFI [35] and ER [6] get down to addressing the domain-shift problem by involving more semantic information, thus consuming extra resources. In particular, MUFI [35] augments semantics by training multi-stream models on multiple datasets. ER [6] expands class names by annotating amount of augmented words crawled from the website. Freeing complex models or additional annotations, we will design a compact model that preserves maximal semantic info of existing classes while synthesizing features of unseen classes.

To tackle the two problems with one stone, we present an end-to-end framework that jointly preserves alignment and uniformity properties for representations on both seen and unseen classes. Here, alignment ensures closeness of visual-semantic features; uniformity encourages the features to distribute uniformly (maximal-info preserving), which improves the possibility that unseen features stand around seen features, mitigating the domain-shift implicitly. Specifically, we formulate a supervised contrastive loss as a combination of two separate terms: one regularizes alignment of features within classes, and the other guides uniformity between semantic clusters. To alleviate the domain-shift explicitly, we generate new features of unseen synthetic classes by our class generator that interpolates and extrapolates features of seen classes. In addition, we introduce closeness and dispersion scores to quantify the two properties and provide new measurements of model generalizability. Fig. 1 illustrates the representations of our method and the SoTA alternative [3]. We train the two models on ten classes sampled from Kinetics-700 [5] and map features on 3D hyperspheres. We observe that our representation shows better closeness within classes and preserves more dispersion between semantic clusters. Experiments validate that our method significantly outperforms SoTA by relative improvements of 28.1% on UCF101 and 27.0% on HMDB51.

2. Related Work

Supervised video classification (SVC): SVC tackles general classes initially (*e.g.*, YouTube-8M dataset [1]), then specific to action recognition recently (*e.g.*, large-scale Kinetics-700 dataset [5]). Learning temporal features is the main task of SVC. In the beginning, video features are generated via NetVLAD [23, 29] that fuses static features of multiple frames. Then, temporal/motion features of videos are optimized directly. We categorize the methods into two-stream 2D-CNN and 3D-CNN based. Two-stream models [22, 38, 46] extract spatial and temporal features by performing separate 2D-CNN modules. [17, 25] extracts motion features by computing 2D-CNN features' difference between neighboring frames. Furthermore, [42] proposes C3D to fuse spatial and temporal features via an independent 3D-CNN module. Even C3D helps achieve promising results [10, 41], its large parameters bring burdens to model optimization. Instead, I3D [5] and P3D [34] design 3D-CNN-like modules by combining 1D temporal and 2D spatial filters. Recently, a more efficient R(2+1)D module [43] has been widely used, which includes a pseudo-3D kernel (2D spatial + 1D temporal) in residual networks. In this paper, we apply our model in action recognition and perform R(2+1)D for better spatial-temporal feature extraction.

Zero-shot video classification (ZSVC): Existing ZSVC methods align visual and semantic features on a unified representation and hope the alignment can be generalized to unseen classes. Most methods design various frameworks to optimize the alignment. Similar to zero-shot image classification [49], some methods [12, 16, 19, 24, 28] learn video attributes first, then design stage-wise framework. Given input videos, [12, 24] learn classifiers for video attributes, then compare the predicted attributes and final class names. However, they cost intensive annotations of video attributes. Instead, [16, 19, 28] utilize pre-trained object detectors to determine object-level class names, then compute similarities between object-level and final class names. Recent work directly computes the similarity between visual and semantic features, and their contributions focus on enhancing visual features. URL [54], TARN [2] and Action2Vec [14] extract spatial-temporal features using a pre-trained C3D and then improve the features via attention modules. The latest model [3] learns visual embeddings by an efficient R(2+1)D module and achieves SoTA results. However, the above methods are not true end-to-end (e2e) models because those utilize Word2vec [30] to extract semantic features. We will justify that lacking e2e learning weakens the alignment since fixed visual/semantic features will bring obstacles to adjusting one to approach another. Except for the above designs, MSE loss [3], ranking loss [14], and center loss [13] are commonly used to regularize the alignment of features. In this paper, we propose a true e2e framework and formulate a supervised contrastive loss, which first considers both alignment and uniformity properties in ZSVC.

Representation learning: In self-supervised and zero-shot learning, representation learning learns features of observed data, which can extract helpful info when applied to downstream tasks. In self-supervised learning, given pairs of positive and negative images/videos, contrastive learning regularizes representations where positives stand close while negatives keep apart. The pioneering work, SimCLR [7] utilizes data augmentation to generate positive instances and maintains a large batch for choosing relatively enough negatives. [33] applies SimCLR to the video domain. To save memory, MoCo [15] presents momentum update to cache a large number of negative instances, then [20, 31] ex-

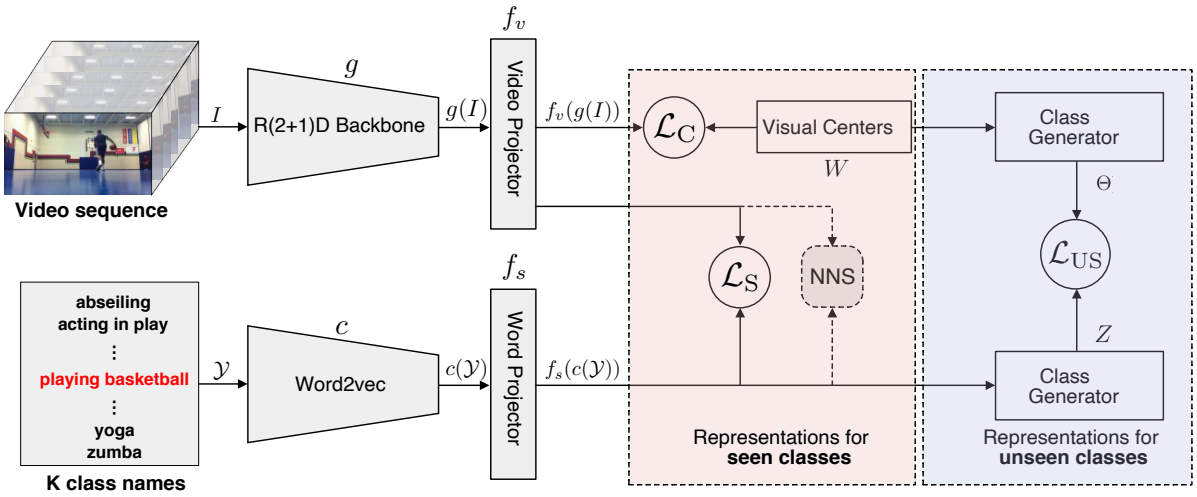


Figure 2. **Architecture of AURL:** From left to right, we map a video sequence I and the class name set $\mathcal{Y}$ to a unified representation $(f_v(g(I)),\ f_s(c(\mathcal{Y})))$. During training, to learn representations of seen classes, we introduce $\mathcal{L}_S$ to preserve *alignment* and *uniformity* properties. For synthetic unseen classes, we introduce $\mathcal{L}_{US}$ to learn the two properties on synthetic visual-semantic features (Θ, Z). To synthesize features of unseen classes, we first utilize $\mathcal{L}_C$ to learn *visual centers* W, then propose *Class Generator* to transform W and existing semantics $f_s(c(\mathcal{Y}))$ into the representation (Θ, Z). During inference, we perform an NNS strategy to obtain the final class.

tends MoCo to video understanding. [53] introduces feature transformation on existing samples to obtain broader and diversified positives and negatives, thus enhancing discrimination. However, the above models learn instance pairs from the same domain, *e.g.*, images or videos. Instead, CLIP [36] exploits features of two domains (*i.e.*, images and texts) guided by a contrastive loss. Motivated by CLIP, the latest models in ZSVC, MUFI [35] and ER [6], extend the self-supervised contrastive loss to a fully-supervised loss that contrasts visual and semantic features. However, MUFI and ER neglect the difference and similarities of the self-supervised and supervised contrastive losses. Considering alignment and uniformity properties, we build connections between the two losses and analyze the advantages of the supervised loss. Besides, MUFI and ER both require extra resources to improve model generalizability. We propose a compact model using a class generator to explicitly synthesize new features of synthetic unseen classes.

3. Alignment-Uniformity aware Representation Learning (AURL)

This section describes Alignment-Uniformity aware Representation Learning (AURL) involving a unified architecture, loss functions, class generator, and two novel metrics, followed by its training and inference strategy, and then discusses similarities and differences against alternatives.

3.1. Architecture

Fig. 2 shows the AURL architecture. Given complete K class names $\mathcal{Y} = \{y_1, \ldots, y_K\}$, and an input video I of class $y_i \in \mathcal{Y}$ (*e.g.*, playing basketball), we by end-to-end learn visual and semantic embeddings. We introduce R(2+1)D [43] as the backbone to generate visual features $g(I)$, and utilize a video projector f_v to implement 3-layer MLP projection (2 fc+bn+ReLU and 1 fc+bn), thus obtain visual embeddings $f_v(g(I)) \in \mathbb{R}^d$. Parallelly, we perform Word2vec c [30] to extract the initial word embeddings $c(\mathcal{Y})$, then learn semantic embeddings $f_s(c(\mathcal{Y})) \in \mathbb{R}^{K\times d}$ by a word projector f_s that has one fc (#node=512) and the 3-layer MLP projection. For convenience, we note $f_v(g(I))$ and $f_s(c(y_i))$ of i-th class as v_{y_i} and s_{y_i} for the below discussions. Compared with SoTA methods [3, 35] that only learn video parts, our AURL end-to-end trains the backbone, video and word projectors, providing more feature flexibility under the regularization of loss functions.

3.2. Alignment-uniformity aware Loss

Can alignment and uniformity properties be preserved in supervised contrastive loss? For self-supervised learning, [8] claims that contrastive loss [7, 33] (see Eq. 1 and *supplementary*) preserves alignment and uniformity properties. Alignment indicates that positive samples should be mapped to nearby features and thus be invariant to unneeded noises. Uniformity [47] means feature vectors should be roughly uniformly distributed on the unit hypersphere, thus bringing better generalization to downstream tasks.

$$\mathcal{L}^{self} = -\log\left[\frac{\exp\left[\lambda \text{sim}(f, f_+)\right]}{\sum_{f_- \in \mathcal{N}} \exp\left[\lambda \text{sim}(f, f_-)\right]}\right]. \quad (1)$$

Here, $(f, f_+), (f, f_-)$ are positive and negative pairs of images/videos, $\mathcal{N}$ is negative set; sim means a similarity function (thus in [-1, +1]); and λ is a temperature parameter.

Closeness in alignment and maximal-info preserving in uniformity are also essential properties of the unified representation learning in ZSVC. However, existing work mainly focuses on the alignment of visual-semantic features [2, 3, 14, 54], the uniformity that improves generalization has not been discussed yet. Here, by leveraging video labels, we formulate a supervised contrastive loss as the combination of alignment and uniformity terms:

$$\mathcal{L}^{sup} = -\log[\frac{\exp[\lambda \text{sim}(v_{y_i}, s_{y_i})]}{\sum_{y_j \in \mathcal{Y}} \exp[\lambda \text{sim}(v_{y_i}, s_{y_j})]}], \tag{2}$$

$$= \lambda \text{SP}_\lambda[\underbrace{-\text{sim}(v_{y_i}, s_{y_i})}_{\text{alignment}} + \frac{1}{\lambda} \underbrace{\text{LSE}(\lambda \text{sim}(v_{y_i}, s_{y_j})_{y_j \in \mathcal{Y} \setminus y_i})}_{\text{uniformity}}],$$

$$\text{where,} \quad \text{SP}_\lambda(x) = \frac{1}{\lambda}\log(1+\exp(\lambda x)),$$

$$\text{LSE}(x) = \log(\sum_{x \in \mathcal{X}} \exp(x)).$$

v_{y_i} and s_{y_i} are visual and semantic features of class y_i, and the complete class set is $\mathcal{Y}$. SP_λ means the SoftPlus function and LSE is LogSumExp. Since Eq. 2 favors $\text{sim}(v_{y_i}, s_{y_i})$ larger, visual and semantic features of the same class will be aligned. The uniformity term tends to maximize the distances between features of different classes using a LogSumExp function, thus spreading features as much as possible. To sum up, our $\mathcal{L}^{sup}$ preserves the alignment and uniformity properties simultaneously.

$\mathcal{L}^{sup}$ **performs better:** Comparing $\mathcal{L}^{sup}$ with $\mathcal{L}^{self}$, we observe that $\mathcal{L}^{sup}$ includes positive pair $\text{sim}(v_{y_i}, s_{y_i})$ in the denominator. Even recent work MUFI [35] and ER [6] also utilize supervised contrastive loss, not only do they neglect the alignment and uniformity properties, but also miss the similarity and difference between the two losses. Here, we show that $\mathcal{L}^{sup}$ maintains advantages of both $\mathcal{L}^{self}$ and triplet loss [37]. We derive upper bounds of $\mathcal{L}^{sup}$ and $\mathcal{L}^{self}$ as follows (the full derivation in *supplementary*):

$$\mathcal{L}^{self} \leq \lambda(\text{sim}_{\text{max}} - \text{sim}(v_{y_i}, s_{y_i}) + \frac{\log(K-1)}{\lambda}), \tag{3}$$

$$\mathcal{L}^{sup} \leq \lambda \max[\text{sim}_{\text{max}} - \text{sim}(v_{y_i}, s_{y_i}) + \frac{\log(K-1)}{\lambda}, 0] + \log(2),$$

where K is the number of classes, sim_{max} is the maximal similarity among all negative pairs ($\text{sim}_{\text{max}} = \max_{y_j \in \mathcal{Y} \setminus y} \text{sim}(v_{y_i}, s_{y_j})$). For a fair comparison, we also reformulate $\mathcal{L}^{self}$ with class labels and obtain its upper bound in Eq. 3. With the upper bounds, we summarize the advantages of $\mathcal{L}^{sup}$ as follows:

1. When $\frac{\log(K-1)}{\lambda} \geq 2$, the two upper bounds will be similar. At this time, $\mathcal{L}^{sup}$ performs as well as $\mathcal{L}^{self}$ in a representation learning task.
2. When $0 \leq \frac{\log(K-1)}{\lambda} < 2$, the upper bound of $\mathcal{L}^{sup}$ has a similar form as triplet loss [37] that facilitates intrinsic ability to perform hard positive/negative mining.
3. $\mathcal{L}^{sup}$ preserves the summation over all negatives in the denominator, thus improving discrimination among classes [39], which has the same motivation with contrastive learning that makes the embedding distribution uniform by increasing the number of negatives [7].

In this paper, we take advantage of $\mathcal{L}^{sup}$ to regularize the representations of both seen and synthetic unseen classes.

$\mathcal{L}^{sup}$ **for seen and unseen classes:** We learn $\mathcal{L}^{sup}$ for both seen and unseen classes (see $\mathcal{L}_{\text{contrast}}$ in Eq. 4 where we utilize cos as a cosine function and map features on the hypersphere.). Specifically, we learn $\mathcal{L}_{\text{S}}$ on visual-semantic features (*i.e.*, v_{y_i} and $s_{y_i} \in \mathbb{R}^{1\times d}, y_i \in \mathcal{Y}$) for seen classes set $\mathcal{Y}$. From the formulation of $\mathcal{L}_{\text{S}}$, we jointly align features of the same class and introduce uniformity that encourages semantic clusters to spread as much as possible, improving the possibility that features of unseen classes fall around existing ones. To offer effective positive/negative visual and semantic pairs that enhance the feature embedding [53], we propose a *class generator* to generate visual and semantic features of synthetic classes $\mathcal{U}$, which are considered as "unseen classes" in comparison with seen classes $\mathcal{Y}$. To retain the *alignment-uniformity* properties, we utilize $\mathcal{L}_{\text{US}}$ to regularize visual and semantic features of K_u unseen classes (*i.e.*, the synthetic features Θ and $Z \in \mathbb{R}^{K_u \times d}$).

$$\mathcal{L}_{\text{contrast}} = \mathcal{L}_{\text{S}} + \mathcal{L}_{\text{US}} \tag{4}$$

$$= -\log[\frac{\exp[\lambda \cos(v_{y_i}, s_{y_i})]}{\sum_{y_j \in \mathcal{Y}} \exp[\lambda \cos(v_{y_i}, s_{y_j})]}] +$$

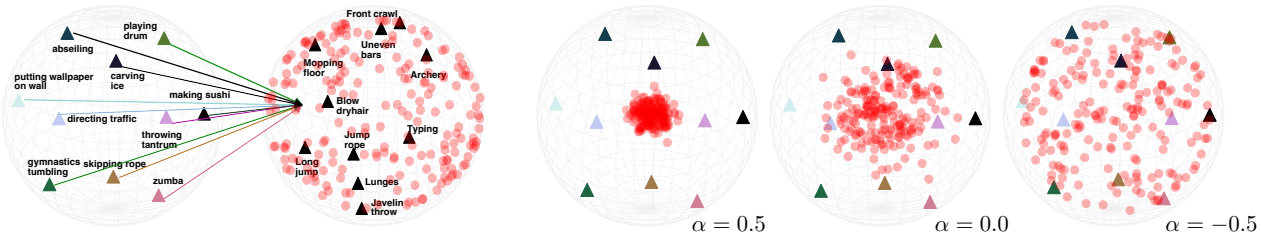
$$\frac{1}{K_u} \sum_{u_i \in \mathcal{U}} -\log[\frac{\exp[\lambda \cos(\Theta_{u_i}, Z_{u_i})]}{\sum_{u_j \in \mathcal{U}} \exp[\lambda \cos(\Theta_{u_i}, Z_{u_j})]}].$$

3.3. Class Generator

To synthesize visual and semantic pairs (Θ, Z), we propose a class generator that applies a uniformly sampled linear transformation to all pairs of visual/semantic features of seen classes. Especially, instead of using single visual feature as the transformed features, we select representative "visual centers" learned from a supervised video classifier that interprets the parameter matrix of fc layer as the centers, as commonly used in [9, 32, 44, 45]. We propose to exploit $\mathcal{L}_{\text{C}}$ as the classification loss, which is an angular softmax loss, helping push all visual features towards their visual centers on the unit hyperspher [48].

$$\mathcal{L}_{\text{C}} = -\log \frac{\exp[\lambda \cos(v_{y_i}, w_{y_i})]}{\sum_{y_j \in \mathcal{Y}} \exp[\lambda \cos(v_{y_i}, w_{y_j})]}. \tag{5}$$

Here, v_{y_i} and w_{y_i} indicate a single visual feature and the learned visual center, respectively. With the visual centers w_{y_i} and the corresponding semantic features s_{y_i}, we inter- and extra-polates (*i.e.*, linearly combine) these features to



(a) Synthetic classes can cover test classes. **(b) Generate unseen classes w/ different parameters.**

Figure 3. **Illustration of Class Generator:** (a) Synthetic classes • are generated by linearly combining features of seen classes (△ with colors); ▲ means test classes. (b) Feature transformation w/ different α synthesizes semantics covering various size of regions.

fill in incomplete class points on the hypersphere:

$$\Theta = M \times \text{norm}(W), \tag{6}$$
$$Z = M \times \text{norm}(s_{\mathcal{Y}}),$$
$$\text{norm}(W) = [\frac{w_{y_1}}{\|w_{y_1}\|}, \ldots, \frac{w_{y_D}}{\|w_{y_D}\|}],$$
$$\text{norm}(s_{\mathcal{Y}}) = [\frac{s_{y_1}}{\|s_{y_1}\|}, \ldots, \frac{s_{y_D}}{\|s_{y_D}\|}].$$

Here, Θ and $Z \in \mathbb{R}^{K_u \times d}$ separately represent the synthetic visual centers and semantic features of unseen classes; K_u represents the number of unseen classes and d is the feature dimension. Besides, we apply normalization on W and $s_{\mathcal{Y}}$ (both are $D \times d$ matrix) because learning on a unit hypersphere helps model optimization [47]; D is a hyper-parameter that means how many classes are sampled for unseen-class generator. The matrix $M \in \mathbb{R}^{K_u \times D}$ is used for inter- and extra-polations, whose elements are randomly sampled from a uniform distribution $U(\alpha, 1)$, and $-1 \leq \alpha < 1$. α is another hyper-parameter that controls the distributed range of the synthetic points.

It is worth noting that the settings of hyper-parameters D and α are non-trivial. For D, we prefer $D \geq d$, *i.e.*, the number of seen classes should be larger than the dimension of a hypersphere. Because for a full rank matrix W, a linear combination of the column vector of W can express any vector on the transformed space. We aim to generate as diverse unseen classes as possible to improve the possibility that the synthesized points can cover the classes in the test set. Thus, in experiments, we will select features of *all* seen classes for feature transformation. For α, we choose positive values for interpolation where the synthetic clusters locate inside of seen points (see Fig. 3(b)), and gradually enlarge the cluster regions by decreasing α where the negative value is for extrapolation. Fig. 3 (a) illustrates our *Class Generator* with $D = 10, d = 3, \alpha = -1$ on the Kinetics-700 dataset [18]. We can see our transformation not only provides unseen classes but also approaches test classes (*e.g.*, UCF101 dataset [40]).

3.4. Closeness and Dispersion

To quantify the alignment and uniformity, we introduce two metrics: closeness and dispersion. Closeness measures the mean distance of features within the same class, reflecting the alignment of visual and semantic features.

$$\text{Closeness} = \frac{1}{K} \sum_{y_i \in \mathcal{Y}} [\frac{1}{N_{y_i}} \sum_{n=1}^{N_{y_i}} (1 - \cos(v_{y_i}^n, s_{y_i}^n))], \tag{7}$$

where, N_{y_i} is the # of training videos of class y_i. Besides, to evaluate the uniformity/separation of semantic clusters, we adopt minimal distances among all clusters to compute dispersion. Here, we consider all visual features within the same class as a semantic cluster instead of using one single semantic vector s_{y_i}. For example, $\bar{v}_{y_i}$ is the mean of visual features of the class y_i, and indicates one semantic cluster.

$$\text{Dispersion} = \frac{1}{K} \sum_{y_i \in \mathcal{Y}} \min_{y_k \in \mathcal{Y} \setminus y_i} (1 - \cos(\bar{v}_{y_i}, \bar{v}_{y_k})). \tag{8}$$

The experiments in Sec. 4.2 show that models tested with higher accuracy preserve the lower closeness and higher dispersion in representations. We conclude our two metrics can serve as new measurements of model generalizability.

3.5. Training & Inference

Training: We end-to-end train visual and semantic features and jointly learn the contrastive loss $\mathcal{L}_{\text{contrast}}$ and classification loss $\mathcal{L}_{\text{C}}$, thus obtaining the following overall loss:

$$\mathcal{L}_{\text{AURL}} = \mathcal{L}_{\text{S}} + \mathcal{L}_{\text{US}} + \mathcal{L}_{\text{C}}. \tag{9}$$

We will justify our end-to-end training is critical for alignment and uniformity properties, and validate our compact model with $\mathcal{L}_{\text{AURL}}$ outperforms SoTA alternatives.

Inference: we train AURL on source dataset $\mathcal{I}$ with K seen classes $\mathcal{Y} = \{y_1, \ldots, y_K\}$, and evaluate the model on target dataset $\mathcal{I}^t$ with T unseen classes $\mathcal{Y}^t = \{y_1^t, \ldots, y_T^t\}$. In this paper, we follow the strict problem setting in [3], which requires training classes $\mathcal{Y}$ have no overlap with test classes $\mathcal{Y}^t$. Mathematically, we re-write the requirement as:

$$\forall y \in \mathcal{Y}, \ \min_{y^t \in \mathcal{Y}^t} (1 - \cos(c_y, c_{y^t})) > \tau, \tag{10}$$

where c_i means Word2vec features of class i, τ is the distance threshold. We utilize the Nearest Neighbor Search

Table 1. Comparisons between AURL and alternative methods.

Methods	ET	AUL	ERF	UCG	SLR
SoTA [3]	×	×	✓	×	✓
MUFI [35]	×	×	×	×	×
ER [6]	✓	×	×	×	×
AURL (ours)	✓	✓	✓	✓	✓

*__ET__: end-to-end trainable, __AUL__: alignment-uniformity learning, __ERF__: extra resources free, __UCG__: unseen class generator, __SLR__: strict label requirement.

(NNS) strategy to obtain the final label of query video I^t:

$$\underset{y^t \in \mathcal{Y}^t}{\operatorname{argmax}} \cos(f_v(g(I^t)), s_{y^t}). \tag{11}$$

3.6. Comparisons with Related Work

The closest studies to our AURL are SoTA [3], MUFI [35], and ER [6]. Table 1 summarizes their similarities and differences. SoTA only utilizes MSE loss to regularize feature alignment within seen classes, limiting model generalizability. MUFI and ER both implicitly increase semantic info to improve the generalization. MUFI trains multi-stream models across multiple datasets. ER crawls and annotates a number of web words to expand existing class names. Unlike that MUFI and ER both require extra resources, our AURL is a compact model to utilize the uniformity that helps preserve maximal info of existing features, and introduce a class generator to synthesize more semantics explicitly. Even MUFI and ER adopt the supervised contrastive loss, they neglect how alignment and uniformity properties affect ZSVC. Besides, our AURL follows the strict label requirement (in Eq. 10) that classes of training and test sets are far away from each other, which manifests the nature of ZSVC. At last, compared with ER that only trains the last fc layers, AURL utilizes a true e2e training strategy that is critical to realize the two properties.

4. Experiments

4.1. Settings

Datasets: We train our AURL on the Kinetics-700 dataset [18] and evaluate it on UCF101 [40] and HMDB51 [21] datasets. The Kinetics-700 provides download links of YouTube videos annotated with 700 categories of human actions. We collect 555,774 videos using these links. The UCF101 contains 13,320 videos with 101 actions and the HMDB51 has 6,767 videos annotated with 51 actions.

Training protocol: For fair comparisons with the SoTA [3], we select the training videos in Kinetics-700 whose classes have non-overlap with UCF101 and HMDB51 as described in Eq. 10, and set the same $\tau = 0.05$, thus obtaining 662 classes. AURL is inductive zero-shot learning, thus does not include any test data during training.

Evaluation protocol: Existing ZSVC methods adopt various evaluation protocols to report experimental results. For complete comparisons, we perform three protocols: 1, 3, and N test splits. 1 *test split* reports an accuracy on all videos of UCF101 or HMDB51 set. 3 *test splits* reports an average accuracy by averaging 3 accuracies that are separately evaluated on 3 test sets provided by the UCF101 or HMDB51. N *test splits* also reports the average by averaging N accuracies that are obtained by running N (10 in our method) times testing, in each, m classes are randomly selected (m=50 for UCF101 and m=25 for HMDB51).

Implementation details: We adopt one or multiple *video clips* as one input video of models. We follow the same SoTA settings [3] (*i.e.*, 1 or 25 video clips and 16 frames/clip with size of $1 \times 16 \times 112 \times 112 \times 3$) for fair comparisons. If multiple video clips are used, we take the mean of multiple visual embeddings as the representative embedding. Besides, if a class name contains multiple words, we average the corresponding Word2vec features to represent the class prototype. For the AURL architecture, we set feature dimension of the R(2+1)D backbone as 512 (*i.e.*, $g(I) \in \mathbb{R}^{512}$) and dimension of Word2vec as 300 (*i.e.*, $c(\mathcal{Y}) \in \mathbb{R}^{K \times 300}$), and set the number of nodes in 3-layer MLP of the projector as 2048, 2048, and 2048 separately. During training, we empirically set K_u as 662, λ as 10, D as 662, and α as 0. We deploy the training on 8 Nvidia Tesla V100 GPUs. We set batch size as 256 and synchronize all batch normalization across GPUs following [4, 7]. We implement experiments using PyTorch and Horovod. SGD is our optimizer and a learning rate of 0.05 with a cosine decay schedule [7, 26] is adopted. Then, we set the weight decay as 0.0001 and the SGD momentum as 0.9. The number of training iterations is 58,500 which takes 45 hours.

4.2. Ablation Study

To analyze AURL, we performed extensive ablations that were trained on the Kinetics-700 and evaluated on UCF101 and HMDB51 using 1 *video clip* and 1 *test split* of evaluation protocols. Table 2 summarizes the quantitative results. Fig. 4 visualizes the visual-semantic representation of ablations by sampling 10 classes from Kinetics-700 dataset and setting the features as 3-D for better visualization. We will justify: (1) our model that preserves alignment-uniformity properties performs better than the SoTA method [3] that focuses on alignment only; (2) end-to-end (e2e) training is critical to realize the two properties; (3) our AURL involving the class generator performs the best. From the justifications, we will show that our closeness and dispersion metrics can serve as new measurements of model generalizability. Here, we take the architecture of the SoTA [3] as our **Base** model (*i.e.*, base backbone + fc only for video parts).

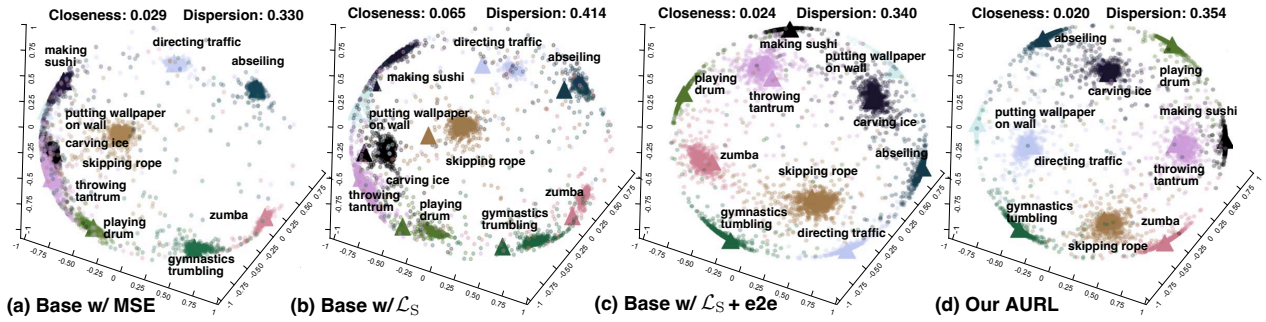


Figure 4. **Ablations:** The representations of ablations w/ MSE loss, our $\mathcal{L}_S$, e2e training, and the AURL. • and △ represent visual and semantic features separately; colors are for different classes. Here, we randomly sample 10 classes from Kinetics-700 for visualization.

Per (1), we compare **Base w/ MSE** (*i.e.*, the SoTA) and **Base w/ $\mathcal{L}_S$**. From the accuracy in Table 2, **Base w/ $\mathcal{L}_S$** largely improves the results by (14.8%, 35.1→40.3) on UCF101 and by (13.6%, 21.3→24.2) on HMDB. Comparing the learned representation in Fig. 4 (a) **w/ MSE** and (b) **w/ $\mathcal{L}_S$** , we observe the semantic clusters of (b) spread more than (a), but the alignment within classes gets worse, for example, visual and semantic features are not calibrated for classes "skipping rope" and "abseiling". Similarly, we find the same trend in closeness and dispersion metrics shown in Fig. 4 and Table 2, where closeness gets worse (0.029→0.065, 0.30→0.45) but dispersion becomes much better (0.330→0.414, 0.09→0.29). We can see our **Base w/ $\mathcal{L}_S$** presesrving higher uniformity in the trained representation can achieve better generalization when making inference on the test set, even scarify a little alignment.

Per (2), we involve **e2e** training strategy (*i.e.*, base backbone + video projector for video parts; word projector for semantic parts) to the **Base w/ $\mathcal{L}_S$**, and get the **Base w/ $\mathcal{L}_S$ + e2e**, which further improves the accuracy from 40.3 to 43.2 on UCF-101 and from 24.2 to 26.2 on HMDB. Not surprisingly, we observe the alignment is tuned better and uniformity is maintained in good quality, thus obtaining a better trade-off. Referring to Fig. 4 (b) and (c), we see **Base w/ $\mathcal{L}_S$ + e2e** encourages better uniformity that semantic clusters are relatively distributed uniformly across the hypersphere while achieves a satisfying alignment that visual and semantic features are apparently aligned (see classes "skipping rope" and "abseiling" again for comparisons). The similar trends also occur in closeness and dispersion metrics, *i.e.*, (0.024 and 0.340; 0.30 and 0.29) in Fig. 4 and Table 2. We conclude that e2e is critical for adjusting features to meet the regularizations of alignment and uniformity.

Per (3), we apply $\mathcal{L}_{US}$ to unseen classes coupling with the class generator (**CG**), *i.e.*, our **AURL**. Compared **AURL** with **Base w/ $\mathcal{L}_S$ + e2e**, **AURL** steadily improves 2.8% on UCF and 4.6% on HMDB, achieving the best accuracy. Quantitatively, closeness and dispersion reach the best scores, such as (0.29, 0.32) in Table 2 and (0.020, 0.354) in Fig. 4. From the representation of Fig. 4 (d), we see the semantic clusters cover most regions of the hypersphere, which improves the possibility that unseen features fall around existing points, thus bringing a better generalization. Furthermore, we remove **CG** from AURL (*i.e.*, **AURL w/o CG**) to validate the effectiveness of the class generator. Comparing **AURL** and **AURL w/o CG**, we find that the performances of **AURL w/o CG** on UCF and HMDB both decrease, and the accuracy on HMDB even degrades lower than **Base w/ $\mathcal{L}_S$ + e2e**. Thus, we conclude the **CG** is a critical module to enhance the generalization.

Table 2. Ablations of our modules using 1 video clip under the *1 test split* protocol (τ=0.05). Red numbers indicate the best. Closeness↓ better, dispersion↑ better, and top-1 accuracy ↑ better.

Method	$\mathcal{L}_S$	e2e	$\mathcal{L}_{US}$ +CG	Close.	Disper.	UCF top-1	HMDB top-1
Base w/ MSE				0.30	0.09	35.1	21.3
Base w/ $\mathcal{L}_S$	✓			0.45	0.29	40.3	24.2
Base w/ $\mathcal{L}_S$ + e2e	✓	✓		0.30	0.29	43.2	26.2
AURL (ours)	✓	✓	✓	0.29	0.32	44.4	27.4
AURL w/o CG	✓	✓	~~CG~~	0.33	0.32	43.7	25.8

Last but not least, from the above justifications, we summarize that our models consistently improve the accuracy by involving the proposed modules; closeness/dispersion measured on the learned representations have agreements with the accuracy evaluated on test sets, providing model evaluations even prior to making inference.

4.3. Comparisons with the Closest SoTA

The closest SoTA to our AURL is the recent work [3], which utilizes a compact model that achieves the SoTA results even under a strict setting (*i.e.*, Eq. 10). Table 1 summarizes the similarity and difference between the SoTA and our AURL. Table 3 shows the comprehensive comparisons quantitatively. We reported the SoTA results using the same settings and the authors' released code. For comprehensive comparisons, we include various evaluation protocols including Pre-training, Video clips, and Test splits. Pre-training means that SoTA fine-tunes the pre-trained models on the SUN dataset [50]. From the comparisons, we see our

Table 3. Comparisons with the closest SoTA [3] on both UCF and HMDB datasets. Red numbers indicate the best.

Method	Pre-training	Video clips	Test splits	UCF top-1	UCF top-5	HMDB top-1	HMDB top-5
SoTA		1	10	43.0	68.2	27.0	54.4
	✓	1	10	45.6	73.1	28.1	51.8
AURL		1	10	55.1	79.3	34.3	65.1
SoTA		25	10	48.0	74.2	31.2	58.3
	✓	25	10	49.2	77.0	32.6	57.1
AURL		25	10	58.0	82.0	39.0	69.5
SoTA		1	1	35.1	56.4	21.3	42.2
	✓	1	1	36.8	61.7	23.0	41.3
AURL		1	1	44.4	70.0	27.4	53.2
SoTA		25	1	37.6	62.5	26.9	49.8
	✓	25	1	39.8	65.6	27.2	47.4
AURL		25	1	46.8	73.1	31.7	58.9

AURL consistently surpasses the SoTA under each evaluation protocol. Specifically, the smallest improvements happen at (25 Video clips, 1 Test splits) by (17.6, 16.5)% improvements on UCF top-1 and HMDB top-1, and the largest comes at (1 Video clip, 10 Test split) by (28.1, 27.0)% increases on UCF top-1 and HMDB top-1. **To conclude, AURL outperforms the SoTA by a large margin.**

4.4. Comparisons with the Alternatives

Table 4 shows the comparisons with the alternatives. The SoTA [3] and our AURL with $\star$ mean the two methods follow the strict label requirement in Eq.10. From the results, we observe that our AURL surpasses all the alternatives in various challenging situations. Specifically, we summarize the challenges: first, fewer test splits are harder testing situations, *e.g.*, for SAOE [27], 3 *vs.* 10 splits corresponds to 32.8 *vs.* 40.4 on UCF; second, strict label requirement ($\star$) serves more difficult situation, *e.g.*, our AURL* w/ 10 (the more) test splits achieves even worse results than AURL w/ 3 splits; third, some methods acquire extra training datasets (*e.g.*, Kinetics + extra 5 datasets trained in MUFI [35]), additional semantic classes (*e.g.*, web words used in ER [6]), and even training videos sampled from the same domain as the test set (*e.g.*, tr/te are both UCF or HMDB in TARN [2], Act2Vec [14], PSGNN [13], and ER [6]), which provide more difficulties to be competed against for other methods. Correspondingly, we find the superiority of our AURL as below: (1) AURL w/ 1 (the fewest) test split outperforms most methods w/ 3 or 50 splits, *e.g.*, (46.8, 31.7) of AURL *vs.* (36.3, -) of OPCL and (43.0, 32.6) of PSGNN on (UCF, HMDB) dataset; (2) AURL* w/ more strict requirements but w/o extra datasets competes against all the SoTA alternatives, *e.g.*, (58.0, 39.0) of AURL *vs.* (51.8, 35.3) of ER, and (56.3, 31.0) of MUFI on (UCF, HMDB) dataset. **To sum up, our AURL reaches the new SoTA in ZSVC**.

Table 4. Comparisons with SoTA alternatives on both UCF and HMDB datasets. Results of alternatives were obtained from original papers, and the higher, the better. Red and blue numbers indicate the best and second best. $\star$ means using τ=0.05 in Eq.10.

Method	Test splits	Train dataset	UCF top-1	Train dataset	HMDB top-1
SoTA* [3]	1	Kinetics	37.6	Kinetics	26.9
AURL*	1	Kinetics	46.8	Kinetics	31.7
Obj2act [16]	3	-	30.3	-	15.6
SAOE [27]	3	-	32.8	-	-
OPCL [13]	3	-	36.3	-	-
MUFI [35]	3	Kinetics+	56.3	Kinetics+	31.0
AURL	3	Kinetics	60.9	Kinetics	40.4
TARN [2]	30	UCF	23.2	HMDB	19.5
Act2Vec [14]	-	UCF	22.1	HMDB	23.5
SAOE [27]	10	-	40.4	-	-
PSGNN [13]	50	UCF	43.0	HMDB	32.6
OPCL [13]	10	-	47.3	-	-
SoTA* [3]	10	Kinetics	48.0	Kinetics	32.7
DASZL [19]	10	-	48.9	-	-
ER [6]	50	UCF	51.8	HMDB	35.3
AURL*	10	Kinetics	58.0	Kinetics	39.0

Finally, we conduct AURL* with pre-extracted features (trained only on video and word projectors). We observe AURL w/ pre-extracted features achieves comparable performance with the e2e AURL – (59.5, 38.2) *vs.* (58.0, 39.0) on UCF and HMDB. This suggests AURL can achieve high performance without carefully finetuning video features.

Limitations and possible solutions. Even our AURL achieves promising results, there are still two problems to be concerned. (1) Uniformity of visual features within classes could be included to further increase info-preserving, introducing contrastive learning between videos may be a possible solution. (2) It will be helpful to study how the class generator affects the overall optimization during training.

5. Conclusion

This paper learns representation awareness of both alignment and uniformity properties for seen and unseen classes. We reformulate a supervised contrastive loss to jointly align visual-semantic features and encourage semantic clusters to distribute uniformly. To explicitly synthesize features of unseen semantics, we propose a class generator that performs feature transformation on features of seen classes. Besides, we introduce closeness and dispersion to quantify the two properties, providing new measurements for generalization. Extensive ablations justify the effectiveness of each module in our model. Comparisons with the SoTA alternatives validate our model reaches the new SoTA results.

Acknowledgement. Research was supported by Natural Science Foundation of China under grant 62076036.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv:1609.08675*, 2016. 2

[2] Mina Bishay, Georgios Zoumpourlis, and Ioannis Patras. Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition. *arXiv:1907.09021*, 2019. 1, 2, 4, 8

[3] Biagio Brattoli, Joseph Tighe, Fedor Zhdanov, Pietro Perona, and Krzysztof Chalupka. Rethinking zero-shot video classification: End-to-end training for realistic applications. In *CVPR*, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv:2006.09882*, 2020. 6

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 2

[6] Shizhe Chen and Dong Huang. Elaborative rehearsal for zero-shot action recognition. In *ICCV*, 2021. 1, 2, 3, 4, 6, 8

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2, 3, 4, 6

[8] Ting Chen, Calvin Luo, and Lala Li. Intriguing properties of contrastive losses. *arXiv preprint arXiv:2011.02803*, 2020. 3

[9] Cunxiao Du, Zhaozheng Chen, Fuli Feng, Lei Zhu, Tian Gan, and Liqiang Nie. Explicit interaction model towards text classification. In *AAAI*, 2019. 4

[10] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 2

[11] Yanwei Fu, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. Transductive multi-view zero-shot learning. *TPAMI*, pages 2332–2345, 2015. 1

[12] Chuang Gan, Tianbao Yang, and Boqing Gong. Learning attributes equals multi-source domain generalization. In *CVPR*, 2016. 2

[13] Junyu Gao, Tianzhu Zhang, and Changsheng Xu. Learning to model relationships for zero-shot video classification. *TPAMI*, pages 3476–3491, 2020. 1, 2, 8

[14] Meera Hahn, Andrew Silva, and James M Rehg. Action2vec: A crossmodal embedding approach to action learning. *arXiv:1901.00484*, 2019. 1, 2, 4, 8

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2

[16] Mihir Jain, Jan C Van Gemert, Thomas Mensink, and Cees GM Snoek. Objects2action: Classifying and localizing actions without any video example. In *ICCV*, 2015. 1, 2, 8

[17] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *ICCV*, 2019. 2

[18] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv:1705.06950*, 2017. 5, 6

[19] Tae Soo Kim, Jonathan Jones, Michael Peven, Zihao Xiao, Jin Bai, Yi Zhang, Weichao Qiu, Alan Yuille, and Gregory D Hager. Daszl: Dynamic action signatures for zero-shot learning. In *AAAI*, 2021. 1, 2, 8

[20] Haofei Kuang, Yi Zhu, Zhi Zhang, Xinyu Li, Joseph Tighe, Sören Schwertfeger, Cyrill Stachniss, and Mu Li. Video contrastive learning with global context. *arXiv:2108.02722*, 2021. 2

[21] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011. 6

[22] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019. 2

[23] Rongcheng Lin, Jing Xiao, and Jianping Fan. Nextvlad: An efficient neural network to aggregate frame-level features for large-scale video classification. In *ECCV Workshops*, 2018. 2

[24] Jingen Liu, Benjamin Kuipers, and Silvio Savarese. Recognizing human actions by attributes. In *CVPR*, 2011. 1, 2

[25] Zhaoyang Liu, Donghao Luo, Yabiao Wang, Limin Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Tong Lu. Teinet: Towards an efficient architecture for video recognition. In *AAAI*, 2020. 2

[26] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv:1608.03983*, 2016. 6

[27] Pascal Mettes and Cees GM Snoek. Spatial-aware object embeddings for zero-shot localization and classification of actions. In *ICCV*, 2017. 1, 8

[28] Pascal Mettes, William Thong, and Cees GM Snoek. Object priors for classifying and localizing unseen actions. *IJCV*, pages 1954–1971, 2021. 1, 2

[29] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *arXiv:1706.06905*, 2017. 2

[30] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv:1301.3781*, 2013. 2, 3

[31] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *CVPR*, 2021. 2

[32] Ofir Press and Lior Wolf. Using the output embedding to improve language models. *arXiv:1608.05859*, 2016. 4

[33] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *CVPR*, 2021. 2, 3

[34] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017. 2

[35] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Xiao-Ping Zhang, Dong Wu, and Tao Mei. Boosting video representation learning with multi-faceted integration. In *CVPR*, 2021. 1, 2, 3, 4, 6, 8

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv:2103.00020*, 2021. 3

[37] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 4

[38] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 2

[39] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPs*, 2016. 4

[40] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*, 2012. 5, 6

[41] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Gate-shift networks for video action recognition. In *CVPR*, 2020. 2

[42] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 2

[43] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 1, 2, 3

[44] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, pages 926–930, 2018. 4

[45] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, 2017. 4

[46] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 2

[47] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 2020. 3, 5

[48] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *WACV*, 2018. 4

[49] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *TPAMI*, pages 2251–2265, 2018. 2

[50] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 7

[51] Timothy M. Hospedales Xu, Xun and Shaogang Gong. Multi-task zero-shot action recognition with prioritised data augmentation. In *ECCV*, 2016. 1

[52] Chenrui Zhang and Yuxin Peng. Visual data synthesis via gan for zero-shot video classification. In *IJCV*, 2018. 1

[53] Rui Zhu, Bingchen Zhao, Jingen Liu, Zhenglong Sun, and Chang Wen Chen. Improving contrastive learning by visualizing feature transformation. *arXiv:2108.02982*, 2021. 3, 4

[54] Yi Zhu, Yang Long, Yu Guan, Shawn Newsam, and Ling Shao. Towards universal representation for unseen action recognition. In *CVPR*, 2018. 1, 2, 4