Supplementary Material for "Elaborative Rehearsal for Zero-shot Action Recognition"

Shizhe Chen*
Inria, France
shizhe.chen@inria.fr

Dong Huang
Carnegie Mellon University, USA
donghuang@cmu.edu

1. Construction of Elaborative Description

Different from the ImageNet object concepts universally defined in standard dictionaries, there are no standard sources to define action classes. We collect Elaborative Descriptions (ED) for action classes in two steps: firstly automatically crawling candidate sentences to describe action classes from the Internet; then manually selecting or modifying a minimum set of candidate sentences as the EDs. We release the collected EDs publicly[1].

In the first crawling step, we utilize Wikipedia and online dictionaries. Given an action class such as "dumpster diving" as query, we use Wikipedia crawling toolkit[2] to collect summary of the first page returned by Wikipedia. This page is usually useful for describing novel actions such as "photobombing" and collocations such as "clean and jerk". We also let Wikipedia find a relevant page title for the query in case no exact page is matched with the query. But to be noted, the returned page can be noisy, especially for compositional action classes. For example, the query "assembling computer" gets the page "assembly language" in computer science. Therefore, we further crawl dictionary definitions[3] for words and phrases in the query action class. We split crawled data into candidate sentences via spacy toolkit[4], and remove non-ascii letters in each sentence.

In the second cleaning step, we represent candidate sentences and a video exemplar in a webpage to annotators as shown in Figure 1. We ask the annotator to select or modify a minimum set of candidate sentences to describe the action class. If no candidate sentences are qualified, the annotator can write a new definition. It takes less than 20s on average to generate the ED per action class. The average length of EDs for actions in the Kinetics dataset is 36 words.

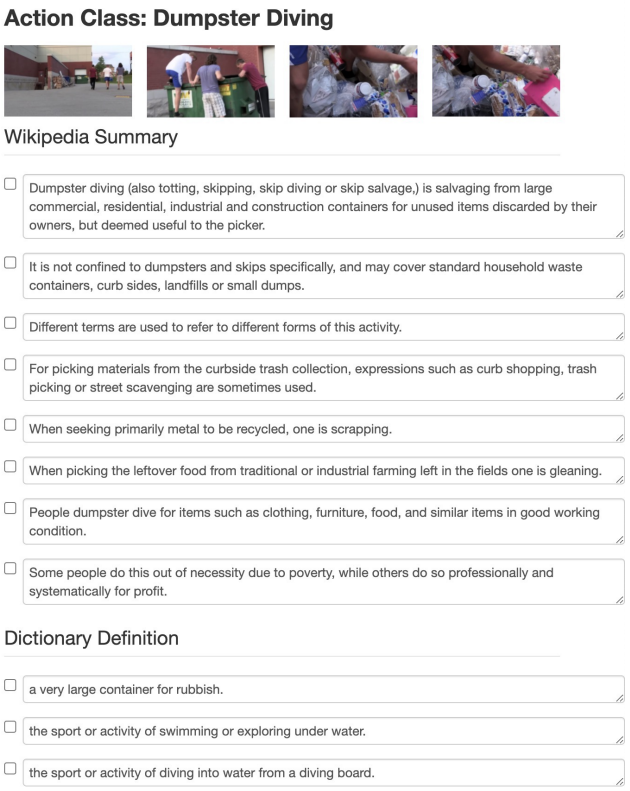


Figure 1: The annotation interface to collect Elaborative Descriptions (ED) for action classes.

2. The Proposed Kinetics ZSAR Benchmark

We use Kinetics-400 [4] as the training dataset and the associated 400 action classes as seen classes. The new classes in Kinetics-600 [3] are used as unseen classes. Due to some renamed, removed or split classes in the evolution from Kinetics-400 to Kinetics-600, it is problematic to obtain new classes by simply selecting action classes that are not in the original (400) action names set. In these ambiguous classes, the videos are still the same even if the class names are different in Kinetics-600. Therefore, we further

*This work was performed when Shizhe Chen was at Carnegie Mellon University.

[1] https://github.com/DeLightCMU/ElaborativeRehearsal/blob/main/datasets/Kinetics/zsl220/classes620_label_defn.json

[2] https://github.com/goldsmith/Wikipedia/

[3] https://dictionaryapi.dev/

[4] https://spacy.io/

	# classes	# videos		
		split1	split2	split3
train	400	212,577	212,577	212,577
val	60	2,670	2,712	2,663
test	160	14,131	14,078	14,167

Table 1: Dataset statistics of our Kinetics ZSAR benchmark.

use the overlapping videos as additional cues to find new classes in Kinetics-600. In this way, we obtained 220 new action classes outside of Kinetics-400.

As mentioned in [11], it is necessary to hold a validation class split that is disjoint from the training and testing classes, to tune hyper-parameters of the zero-shot methods. Therefore, we randomly split the 220 new classes in Kinetics-600 into the 60 validation and 160 testing classes. To avoid the potential bias in only one split, we independently split the classes for three times to improve the robustness of evaluation. The validation and testing videos are from the original Kinetics-600 splits respectively. To be noted, since the training set is the same for the three splits, the ZSAR methods only need to train once on the training set, and then different validation sets are used to select the best models for the respective testing sets. The dataset statistics of the three splits are shown in Table 1.

3. Our Implemented Baseline Models

As there is no baseline to compare on our newly proposed Kinetics ZSAR benchmark, we implement the following state-of-the-art ZSL algorithms: (1) DEVISE [5]; (2) ALE [1]; (3) SJE [2]; (4) DEM [12]; (5) ESZSL [9]; and (6) GCN [6].

Among them, DEVISE, ALE, SJE and ESZSL use bilinear compatibility function to associate video v and class y with different objectives in training:

$$F(v, y; W) = \phi(v)^T W \psi(y) \tag{1}$$

All the methods use the same ST video encoding $\phi(v)$ as ours. The semantic representation $\psi(y)$ for action classes are L2 normalized mean-pooled Glove42b [8] feature of class names, which shows better performance than other word embeddings and sent2vec embeddings [7]. We revisit the core idea of each method below.

DEVISE [5] uses pairwise ranking objective:

$$\sum_{y \in \mathcal{S}} [\Delta(y^n, y) + F(v^n, y; W) - F(v^n, y^n; W)]_+ \tag{2}$$

where $\Delta(y^n, y) = 0$ if $y^n = y$ otherwise 0.2.

ALE [1] uses weighted approximate ranking objective:

$$\sum_{y \in \mathcal{S}} \frac{l_{r_{\Delta(v^n, y^n)}}}{r_{\Delta(v^n, y^n)}} [\Delta(y^n, y) + F(v^n, y; W) - F(v^n, y^n; W)]_+ \tag{3}$$

where $l_k = \sum_{i=1}^{k} \alpha_i$ and $r_{\Delta(v^n, y^n)}$ is defined as:

$$\sum_{y \in \mathcal{S}} \mathbf{1}(F(v^n, y; W) + \Delta(y^n, y) \geq F(v^n, y^n; W)) \tag{4}$$

We use $\alpha_i = 1/i$ which puts a high emphasis on the top of the rank list.

SJE [2] uses hard negative label mining with the training objective as follows:

$$\max_{y \in \mathcal{S}} [\Delta(y^n, y) + F(v^n, y; W) - F(v^n, y^n; W)]_+ \tag{5}$$

DEM [12] uses the visual space as the embedding space, which learns a non-linear mapping from class features to visual features and minimizes the model with MSE loss:

$$\frac{1}{N} \sum_{i=1}^{N} ||\phi(v^n) - f_1(W_2 f_1(W_1 \psi(y^n))||^2 + \lambda(||W_1||^2 + ||W_2||^2) \tag{6}$$

ESZSL [9] applies a square loss to the pairwise ranking formulation and adds regularization terms to optimize:

$$\gamma ||W\psi(y)||^2 + \lambda ||\phi(v)^T W||^2 + \beta ||W||^2 \tag{7}$$

There exists a closed form solution for the objective.

GCN [6] is a very recent ZSAR work which builds knowledge graphs for action classes to predict classification weights as [10]. We use the first type of knowledge graphs as their work, which is built based on similarity of class embeddings. Six GCN layers are used to predict classification weights from the built graph.

4. More Ablation Studies

Multimodal-based Channel Attention. In Table 2, we compare our ER-enhanced models with or without multimodal-based channel attention in video semantic representation encoding in Section 3.3 of our main paper. The comparison shows that the proposed channel attention is beneficial to generate better video semantic representations from the ST and object streams.

ER loss. Table 3 presents additional models (using spatial-temporal and object video representations) trained with or without ER loss. The trend is the same as Table 4c in the main paper. The ER loss improves the generalization ability on unseen actions by 2.6% on Top-1 accuracy and 3.0% on Top-5 accuracy.

model	Top-1 (%)	Top-5 (%)
w/o MCA	41.0 ± 1.7	71.9 ± 0.7
w/ MCA	**42.1** ± **1.4**	**73.1** ± **0.3**

Table 2: Comparison of ER-enhanced models with or without multimodal-based channel attention (MCA) on Kinetics ZSAR benchmark.

Video	ER	top-1	top-5
ST+Obj	w/o	39.5 ± 1.4	70.1 ± 0.6
	w/	**42.1** ± **1.4**	**73.1** ± **0.3**

Table 3: Comparison of ER-enhanced models with or without ER loss on Kinetics ZSAR benchmark.

# objects VE	# objects ER	Top-1 (%)	Top-5 (%)
5	0	39.5 ± 1.4	70.1 ± 0.6
5	1	41.5 ± 1.9	70.9 ± 1.0
5	5	**42.1** ± **1.4**	**73.1** ± **0.3**
5	10	41.0 ± 1.6	72.0 ± 1.2
0	5	37.1 ± 1.7	69.3 ± 0.8
1	5	37.6 ± 1.0	68.9 ± 0.8
5	5	**42.1** ± **1.4**	**73.1** ± **0.3**
10	5	42.0 ± 1.3	72.3 ± 0.6

Table 4: Comparison of ER-enhanced models using different numbers of objects in object stream of video encoding (VE) and ER loss (ER) on our Kinetics ZSAR benchmark.

Model Video	Model Loss	ObjSet	Top-1 (%)	Top-5 (%)
ST	AR	-	31.0 ± 1.2	63.2 ± 0.4
Obj	AR + ER	1K	24.8 ± 0.7	51.7 ± 0.7
Obj	AR + ER	21K	36.7 ± 1.0	63.2 ± 0.5
ST + Obj	AR + ER	1K	34.7 ± 1.1	67.4 ± 1.0
ST + Obj	AR + ER	21K	42.1 ± 1.4	73.1 ± 0.3

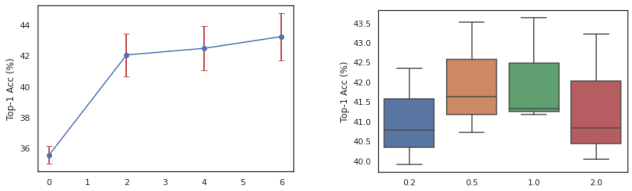
Table 5: Comparison of using different sets of object concepts ("ObjSet") in our ER model on our Kinetics ZSAR benchmark.

Number of Object Concepts. Table 4 presents ZSAR performances using different numbers of object concepts predicted in the object stream of video encoding and ER loss respectively. We can see that the ZSAR performance first increases with the number of objects and then decreases, which might result from incorrectly detected (false positive) object concepts.

Different Object Concepts. We compare different sets of object concepts in Table 4. In our main paper, we use the full concept set in ImageNet21k from the BiT model. We compare it with concepts in ImageNet1k from Resnext50[5] image classification model. The predicted concepts of the latter are not as accurate as the former due to less training data and fewer concept classes. When only using the object concepts as video semantic representation, we can see that ZSAR performance of the ImageNet1k concepts are much worse than that of ImageNet21k and ST features. It indicates that the object concepts set and recognition performance are important. Though objects from ImageNet1k alone are not competitive, they are still complementary to ST video features. The combination of object and ST feature in our full ER model also achieves better performance.

Model Video	Model Loss	Top-1 (%)	Top-5 (%)
ST	AR	31.0 ± 1.2	63.2 ± 0.4
ST (NL)	AR + ER	32.0 ± 0.9	63.9 ± 0.6
ST + Obj	AR + ER	42.1 ± 1.4	73.1 ± 0.3
ST (NL) + Obj	AR + ER	42.7 ± 1.6	73.3 ± 0.6

Table 6: Comparison of using different Spatio-Temporal (ST) features on Kinetics ZSAR benchmark. **NL** denotes non-local.



(a) # of finetuned layers. (b) lambda of ER loss.

Figure 2: Top-1 accuracy for different hyper-parameters.

Different Spatio-Temporal(ST) Features. We further verify the generalization of our approach on different ST features. Table 6 shows the results. We compare the TSM model and an enhanced TSM with non-local attentions for ST feature extraction. Better ST features are beneficial to the ZSAR performance.

Number of Finetuned Layers in BERT Model. As shown in Figure 2a, finetuning more layers in BERT continuously improves the performance, which however consumes more resources, *e.g.* we use 1 RTX 2080Ti to finetune 2 layers in BERT, but need 4 GPUs to finetune 6 layers.

λ for Elaborative Rehearsal Loss. Figure 2b presents the performance of different λs for the ER loss, which suggests that the ER loss is better to set as equal contributions as the action classification loss.

[5] https://pytorch.org/docs/stable/torchvision/models.html#classification

References

[1] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *IEEE transactions on pattern analysis and machine intelligence*, 38(7):1425–1438, 2015. 2

[2] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2927–2936, 2015. 2

[3] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. 1

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1

[5] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013. 2

[6] Pallabi Ghosh, Nirat Saini, Larry S Davis, and Abhinav Shrivastava. All about knowledge graphs for actions. *arXiv preprint arXiv:2008.12432*, 2020. 2

[7] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. In *NAACL-HLT*, 2018. 2

[8] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 2

[9] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, pages 2152–2161, 2015. 2

[10] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6857–6866, 2018. 2

[11] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2018. 2

[12] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2021–2030, 2017. 2