C2C: Component-to-Composition Learning for Zero-Shot Compositional Action Recognition (Supplementary Material)

Rongchang Li[1,2], Zhenhua Feng[1,2,3,4], Tianyang Xu[1], Linze Li[1], Xiao-Jun Wu[1✉], Muhammad Awais[2,4], Sara Atito[2,4], and Josef Kittler[2,3]

[1] School of Artificial Intelligence and Computer Science, Jiangnan University, China
[2] Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, UK
[3] School of Computer Science and Electronic Engineering, University of Surrey, UK
[4] Surrey Institute of People-centred AI (SI-PAI), University of Surrey, UK
{li_rongchang,linze.li}@stu.jiangnan.edu.cn;
feng-zhenhua@outlook.com; {wu_xiaojun,tianyang_xu}@jiangnan.edu.cn;
{muhammad.awais, sara.atito, j.kittler}@surrey.ac.uk;

1 Hilbert-Schmidt Independence Criterion

Hilbert-Schmidt Independence Criterion (HSIC) [5] is a widely used measure of independence. It is defined based on the cross-covariance operator between the distributions in the Reproducing Kernel Hilbert Space (RKHS). Let $X \sim P_X : \Omega \to \mathcal{X}$ and $Y \sim P_Y : \Omega \to \mathcal{Y}$ be the two random variables in the non-empty set X and Y, respectively. They are sampled from P_X and P_Y distribution of the sample space Ω. The formulation of HSIC is defined as:

$$\begin{aligned} \mathrm{HSIC}\left(\mathbb{P}_{XY}, \mathcal{H}, \mathcal{G}\right) &= \left\|C_{XY}\right\|^2 \\ &= \mathbb{E}_{XYX'Y'}\left[k_X\left(X, X'\right) k_{Y'}\left(Y, Y'\right)\right] \\ &+ \mathbb{E}_{XX'}\left[k_X\left(X, X'\right)\right] \mathbb{E}_{Y'}\left[k_Y\left(Y, Y'\right)\right] \\ &- 2\mathbb{E}_{XY}\left[\mathbb{E}_{X'}\left[k_X\left(X, X'\right)\right] \mathbb{E}_{Y'}\left[k_Y\left(Y, Y'\right)\right]\right], \end{aligned} \tag{1}$$

where k_X and k_Y are kernel functions, $\mathcal{H}$ and $\mathcal{G}$ are Hilbert spaces. $\mathbb{E}_{XY}$ is the expectation over X and Y.

In this paper, we adopt the normalized-HSIC (nHSIC) based on canonical correlation analysis [13, 17]. let $\mathcal{D} := \{(x_1, y_1), \ldots, (x_m, y_m)\}$ contains m samples. k_X and k_Y are kernel functions. $\mathbf{K}_X \in \mathbb{R}^{m \times m}$ and $\mathbf{K}_Y \in \mathbb{R}^{m \times m}$ have entries $\mathbf{K}_{Xij} = k\left(\mathbf{x}_i, \mathbf{x}_j\right)$ and $\mathbf{K}_{Yij} = k\left(\mathbf{y}_i, \mathbf{y}_j\right)$. nHSIC is obtained by:

$$\mathrm{nHSIC}(\mathbf{X}, \mathbf{Y}) = \mathrm{tr}\left(\widetilde{\mathbf{K}}_X \widetilde{\mathbf{K}}_Y\right), \tag{2}$$

where $\widetilde{\mathbf{K}}_X = \overline{\mathbf{K}}_X \left(\overline{\mathbf{K}}_X + \epsilon m \mathbf{I}_m\right)^{-1}$ and $\widetilde{\mathbf{K}}_Y = \overline{\mathbf{K}}_Y \left(\overline{\mathbf{K}}_Y + \epsilon m \mathbf{I}_m\right)^{-1}$, $\overline{\mathbf{K}}_X$ and $\overline{\mathbf{K}}_Y$ are centralized kernel matrices, and ϵ is a small constant.

2 Implementation details

Implementation details on Sth-com. In our experiments, we uniformly divide the input video into 8 segments, selecting one frame from each to form the input video clip. The spatial resolution of each frame is resized to 256 $\times$ 256 and then resized to 224 $\times$ 224. The temperature coefficient τ for loss calculations is set at 0.01. When adopting the vision model + word embedding model paradigm, we use TSM-18 [10] or VideoSwin-T [11] as the video encoder. The weights about spatial operations are pre-trained on Imagenet-1k [2]. Note that we do not use Kinetics [7] pre-trained weights as there is a big domain gap between Kinetics and Sth-V2 [4]. We adopt Adam [9] as the optimizer and cosine learning rate schedule [12], with a batch size of 64. We train the model for 50 epochs, including a warm-up phase during the initial three epochs. For the video encoder, the learning rate is set as 5e-4 with a weight decay of 5e-4 for TSM-18, and 2e-4 with a weight decay of 5e-5 for VideoSwin-T. For the other parts, the learning rate is set to 2.5e-4 with a weight decay of 5e-5.

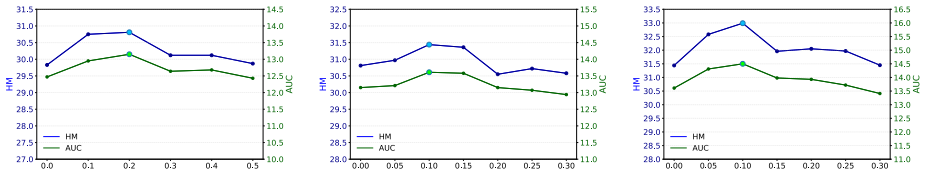
When adopting the vision-language model, we use CLIP [16] for video and label encoding. During the training stage, we only optimize the prompt vectors of the text input, image-to-video adapters borrowed from AIM [18] and the C2C part. We train the model for 30 epochs, including a warm-up phase for the first three epochs. The learning rate of adapters and the C2C part is set as 5e-4 with a weight decay of 1e-4. The learning rate of prompts is set to 1.0e-4 with a weight decay of 1e-5.

Implementation details on C-GQA. To verify the compositional generalization ability of our proposed method, we transfer the proposed Component-to-Composition (C2C) method to the image-based compositional generalization task, i.e., Compositional Zero-shot Learning (CZSL). The most popular benchmarks in CZSL are UT-Zappos [19], MIT-States [6] and C-GQA [14]. However, the UT-Zappos dataset contains a limited number of compositions, and MIT-States suffers from label noise [1]. Therefore we employ the recently introduced C-GQA dataset to evaluate our method. In our experiments, we utilize the updated version of the C-GQA dataset. C-GQA comprises 413 attributes and 674 objects. It includes 5592 compositions in the training set, 1252/1040 (seen/unseen) compositions in the validation set, and 888/923 in the test set. Following CoT [8], we use ViT-B/16 [3], pretrained on ImageNet [2] as our backbone. GloVe [15] is used for word vector embedding. For network training, the batch size is set to 96. The learning rates are set to 1.0e-5 for the visual encoder and 1.0e-4 for the C2C part. The weight decay is set as 5.0e-5. The training epoch is 30, and the learning rate decays by 0.1 at the 20th and 25th epochs.

3 More experimental results

3.1 Loss balancing parameters

We explore the influence of the loss balancing parameters, i.e., α for $\mathcal{L}_{comp}$, β for $\mathcal{L}_{ind}$ and γ for $\mathcal{L}_{new}/\mathcal{L}_{con}$, on the final performance of the trained network.



(a) α **(b)** β **(c)** γ

Fig. 1: Influence of loss balancing parameters on the performance, evaluated in HM (blue) and AUC (green).

We initially set the baseline as only using the composition loss $\mathcal{L}_{com}$. Then we progressively increase α for $\mathcal{L}_{comp}$, beginning at 0 with an interval of 0.1. The results of HM and AUC are shown in Fig. 1a. It achieves peak performance at $\alpha = 0.2$. We then add $\mathcal{L}_{ind}$ and vary β in steps of 0.05, with the results presented in Fig. 1b. It indicates that the optimal result is achieved at $\beta = 0.1$. Last, we explore γ by gradually increasing it from 0 with an interval of 0.05. According to Fig. 1c, it shows that the best performance is achieved at $\gamma = 0.1$. These results demonstrate that choosing appropriate loss balancing parameters enhances HM and AUC, affirming the effectiveness of the proposed independent component learning and enhanced learning strategy.

3.2 Component-specific channel ratio ρ

Table 1: Impact of the component-specific channel ratio in the calculation of L_{ind}. $\rho = 0$ means no independence constrain between $\mathbf{f}_v$ and $\mathbf{f}_o$.

	base	$\rho = 0$	$\rho = 0.25$	$\rho = 0.5$	$\rho = 0.75$	$\rho = 1$
HM	30.81	30.99	31.28	**31.44**	31.20	30.73
AUC	13.15	13.20	13.35	**13.61**	13.39	12.98

In the computation of the loss $\mathcal{L}_{ind}$, we identify certain channels of $\mathbf{f}_v$ and $\mathbf{f}_o$ as component-specific channels, defined as $\mathbf{f}'_v = \mathbf{f}_v[:\rho C], \mathbf{f}'_o = \mathbf{f}_o[:\rho C]$ where $0 \leq \rho \leq 1$. We vary the component-specific channel ratio ρ from 0 to 1 and the results are presented in Tab. 1. $\rho = 0$ means only reducing the general spurious information. It only gains a modest performance improvement. As ρ increases, we observe more considerable performance improvements. However, an excessively high ratio ρ means fewer common feature channels, which may lead to performance degradation. According to Tab. 1, the optimal setting is $\rho = 0.5$.

3.3 Cutmix probability p

Table 2: Impact of the probability of applying CutMix for each training batch.

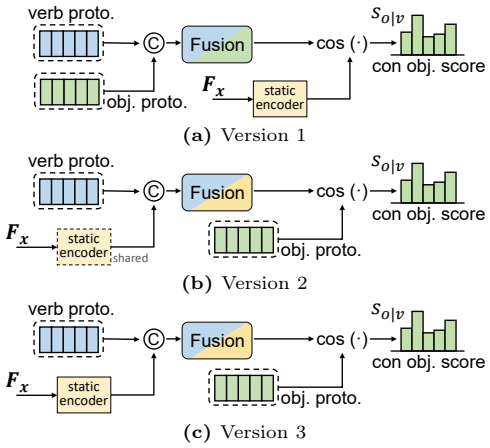
	base	$p = 0.1$	$p = 0.3$	$p = 0.5$	$p = 0.7$	$p = 0.9$
HM	30.81	31.50	32.12	32.40	**32.99**	31.54
AUC	13.16	13.48	14.09	14.28	**14.50**	13.42

Table 3: A comparison between different methods in calculating conditional component scores. The baseline takes verbs and objects completely independent.

	Baseline	Version 1	Version 2	Version 3
HM	29.8	30.2	30.4	**30.8**
AUC	12.2	12.5	12.7	**13.2**

In our proposed enhanced training strategy, we apply CutMix [20] to form novel verb-object compositions. For each training batch, there is a probability p to apply CutMix. We conduct experiments on different p values and report the results in Tab. 2. It shows that the method performs best when we set $p = 0.7$. The results demonstrate that the use of CutMix improves the performance significantly.

3.4 Calculating compatibility score



(a) Version 1

(b) Version 2

(c) Version 3

Fig. 2: Different methods that can be used to calculate the conditional component score. We take the calculation of conditional object score as an example.

In the proposed component-to-composition module, we define the conditional component score as the cosine similarity between the compound prototype-visual features and the prototypes of another component. We explore more methods that can be used for calculating the conditional score, with a comparison presented in Fig. 2. In Fig. 2a, the component prototypes are merged into composition embeddings, with the conditional score derived from the cosine similarity between these embeddings and the visual features. Fig. 2b integrates component prototypes with visual features as the compound prototype-visual features, but the visual encoder shares parameters with that used in the independent component learning module. Fig. 2c is our final scheme where the difference between Fig. 2b is that the visual encoder doesn't share parameters. This approach ensures that the visual features for calculating the conditional score remain flexible and not overly constrained to any specific component.

The above methods are referred to as Version 1, Version 2, and Version 3 in Tab. 3. The baseline approach considers the verb and object completely independent, deriving the composition score by directly multiplying the independent component scores. The results demonstrate that utilizing the conditional component score improves performance. Moreover, the compound prototype-visual features have been proven to be more effective for calculating the conditional component score (Version 2 vs Version 1). Additionally, using visual encoder parameters separately from those in the independent component learning module further improves the performance.

References

1. Atzmon, Y., Kreuk, F., Shalit, U., Chechik, G.: A causal view of compositional zero-shot recognition. Advances in Neural Information Processing Systems **33**, 1462–1473 (2020)
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)
3. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2020)
4. Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al.: The" something something" video database for learning and evaluating visual common sense. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5842–5850 (2017)
5. Gretton, A., Bousquet, O., Smola, A., Schölkopf, B.: Measuring statistical dependence with hilbert-schmidt norms. In: International Conference on Algorithmic Learning Theory. pp. 63–77 (2005)
6. Isola, P., Lim, J.J., Adelson, E.H.: Discovering states and transformations in image collections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1383–1391 (2015)
7. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
8. Kim, H., Lee, J., Park, S., Sohn, K.: Hierarchical visual primitive experts for compositional zero-shot learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5675–5685 (2023)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
10. Lin, J., Gan, C., Han, S.: Tsm: Temporal shift module for efficient video understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7083–7093 (2019)
11. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video swin transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3202–3211 (2022)
12. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. In: International Conference on Learning Representations (2016)

13. Ma, W.D.K., Lewis, J., Kleijn, W.B.: The hsic bottleneck: Deep learning without back-propagation. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 5085–5092 (2020)
14. Naeem, M.F., Xian, Y., Tombari, F., Akata, Z.: Learning graph embeddings for compositional zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 953–962 (2021)
15. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
16. 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: International conference on machine learning. pp. 8748–8763. PMLR (2021)
17. Wang, Z., Jian, T., Masoomi, A., Ioannidis, S., Dy, J.: Revisiting hilbert-schmidt information bottleneck for adversarial robustness. Advances in Neural Information Processing Systems **34**, 586–597 (2021)
18. Yang, T., Zhu, Y., Xie, Y., Zhang, A., Chen, C., Li, M.: Aim: Adapting image models for efficient video action recognition. arXiv preprint arXiv:2302.03024 (2023)
19. Yu, A., Grauman, K.: Fine-grained visual comparisons with local learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 192–199 (2014)
20. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6023–6032 (2019)