Supplementary Material: Temporal Hallucinating for Action Recognition with Few Still Images

Yali Wang[1*] Lei Zhou[1,3*] Yu Qiao[1,2†]
[1] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
[2] The Chinese University of Hong Kong [3] SenseTime Group Limited

1. Gaussian Process

Gaussian process (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution [4]. It is a flexible Bayesian nonparametric prior for data-driven modeling.

Suppose that N input-output pairs, $\{\mathbf{X}, \mathbf{Y}\} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, are generated from

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) + \epsilon, \ \ \epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 I), \tag{1}$$

where we assume that the nonlinear functions $\mathbf{f}(\mathbf{x})$ have a GP prior with kernel $k(\mathbf{x}, \mathbf{x}')$,

$$\mathbf{f}(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')), \tag{2}$$

and the noise ϵ is Gaussian with σ^2.

In this case, the output $\mathbf{y}_*$ of a query input $\mathbf{x}_*$ can be elegantly predicted from $p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y})$, due to the Gaussian property of GP [4],

$$\mathbf{y}_* = \mathbf{k}(\mathbf{x}_*, \mathbf{X})[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}]^{-1}\mathbf{Y}, \tag{3}$$

where the vector $\mathbf{k}(\mathbf{x}_*, \mathbf{X})$ is constructed by computing similarities between query and memory inputs, i.e., $(\mathbf{k})_i = k(\mathbf{x}_*, \mathbf{x}_i)$, $i = 1, ..., N$. The matrix $\mathbf{K}(\mathbf{X}, \mathbf{X})$ is constructed by computing similarities between memory inputs, i.e., $(\mathbf{K})_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$, $i, j = 1, ..., N$. More technical details can be found in [4].

2. Basic Memory Module via GP

In this section, we flexibly adopt GP as a basic memory module in Fig. 1. Specifically, we treat $\{\mathbf{X}, \mathbf{Y}\} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ as input-output pairs in memory. For a query input $\mathbf{x}_*$, GP can predict the query output $\mathbf{y}_*$ from memory, according to Eq. (3).

We rewrite Eq. (3) as a weighted sum of memory outputs,

$$\mathbf{y}_* = \sum\nolimits_{i=1}^{N} w(\mathbf{x}_*, \mathbf{x}_i)\mathbf{y}_i = \mathbf{w}\mathbf{Y}, \tag{4}$$

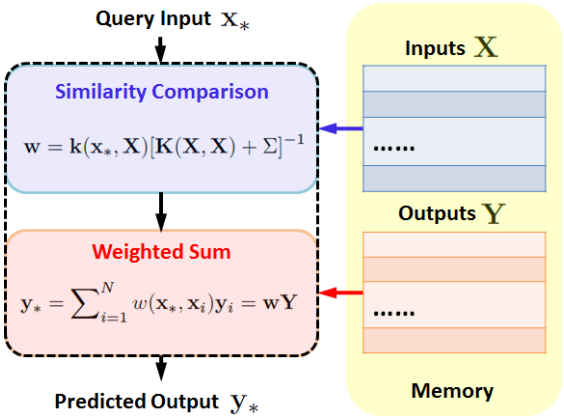

Figure 1. Basic memory module via Gaussian Process (GP).

where the weight vector is encoded by similarity comparison between query and memory,

$$\mathbf{w} = \mathbf{k}(\mathbf{x}_*, \mathbf{X})[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \Sigma]^{-1}. \tag{5}$$

In this case, GP can be used as memory machines, allowing to make effective non-parametric predictions by comparing similarities between query and memory. Furthermore, simple kernel operations in GP may be preferable to encode similarities, compared to large network structure designs with complex training strategies in the previous memory networks [5, 6]. This fact is important to alleviate overfitting, especially when learning with few examples.

3. Data Sets

For WEB101, we use Google Engine to collect images that have the same classes as UCF101. Specifically, we first expand the name of each action class respectively with *man*, *boy*, *woman* and *girl*, aiming at increasing collection diversity. Then, we manually filter out the mislabeled images, noisy images containing only objects or texts, etc.

For DIFF20, the collection procedure is similar to the one of WEB101, except that action classes of DIFF20 are totally different from UCF101. To achieve this, we carefully select 20 action classes from Kinetics [3] and MPII Pose

*Equally-contributed first authors ({yl.wang, lei.zhou}@siat.ac.cn).
†Corresponding author (yu.qiao@siat.ac.cn).

Figure 2. Action categories of DIFF20 data set.

[1]. As shown in Fig. 2, three classes are from MPII Pose (i.e., *playing broomball*, *playing handball*, *playing horn*) and others are from Kinetics. Note that, there exists the class overlap between Kinetics and ActivityNet, e.g., *playing badminton* and *windsurfing*, which we pick from Kinetics, also appear in ActivityNet.

For VOC, we build it from VOC 2012 Action Dataset [2]. It consists of 10 action categories in which 4 categories are overlapped with UCF101 (i.e., Jumping, PlayingInstrument, RidingBike, RidingHorse). To avoid ambiguity of actions from multiple targets, we crop the squared bounding box as one image sample in our VOC. Furthermore, we exclude all samples in the 'other' class of VOC Action 2012 in our experiments, as our main goal is to evaluate if temporal features hallucinated from video memory can boost action recognition with few still images.

4. More Ablation Studies

Performance Convergence. To check when the recognition will saturate, we further report accuracy of WEB101 with 20 / 30 / 40 / 50 training images per category in Table 1. As expected, when the number of training images increases, the accuracy of our models increases and the increasing trend is gradually flat.

No. of Images	20	30	40	50
our TP	62.3	65.9	68.0	69.3
our SP	63.5	66.9	68.5	69.9
our HVM	**63.6**	**67.2**	**68.9**	**70.2**

Table 1. Performance convergence (WEB101). As expected, when the number of training images increases, the accuracy of our models increases and the increasing trend is gradually flat.

Approaches	WEB101	VOC	DIFF20
$\text{TSN}_{imagenet}$	26.4	42.9	66.3
$\text{TSN}_{kinetics}$	32.8	51.4	76.6
Our $\text{HVM}_{kinetics}$	**34.0**	**51.8**	**78.4**

Table 2. Different video memory (the most challenging 1-image case). We use Kinetics [3] to construct our video memory, and compare our HVM with TSN (i.e., its RGB stream, pretrained respectively on ImageNet and Kinetics).

Different Video Memory. We further construct our video memory using Kinetics [3] and report our experimental results. Similar to the construction of UCF101, we randomly select 20 videos from each category of 400 actions in Kinetics. Then, we generate spatial and temporal features (5b layer, 1024 dimension after global pooling) from TSN pretrained on Kinetics, where Kinetics Val Top1 Acc are 69.1 (RGB stream) and 62.1 (Flow stream). We perform HVM for 1-training-image case, and compare it with TSN (i.e., its RGB stream, pretrained respectively on ImageNet and Kinetics). As shown in Table 2, the performance of our approach can be further improved, when using the large-scale Kinetics as video memory.

References

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014.

[2] M. Everingham, L. Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[3] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. In *arXiv:1705.06950*, 2017.

[4] C. E. Rasmussen and C. K. I. Williams. *Gaussian Process for Machine learning*. MIT Press, 2006.

[5] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. End-to-end memory networks. In *NIPS*, 2015.

[6] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *NIPS*, 2016.