Supplemental Material: Learning to Align Sequential Actions in the Wild

Weizhe Liu[1] Bugra Tekin[2] Huseyin Coskun[3] Vibhav Vineet[2] Pascal Fua[4] Marc Pollefeys[2,5]
[1] Tencent AI Lab [2] Microsoft [3] Technische Universität München [4] EPFL [5] ETH Zurich

In the supplemental material, we provide analysis of our approach for fine-grained frame retrieval (Sec. 1) and different training and evaluation strategies (*e.g.* random initialization, single model for all activities, individual models per activity) (Sec. 2- 4). We then present additional ablation studies regarding the hyper-parameters of the model, backbone neural network and the choice of probabilistic distribution for the priors (Sec. 5). Subsequently, we provide implementation details of our approach regarding model architecture and hyperparameters (Sec. 6) and visualize the embedding space learned through our approach (Sec. 7). We finally discuss the limitations and broader potential societal impact of our approach (Sec. 8).

1. Fine-Grained Frame Retrieval

We evaluate our model for the task of fine-grained frame retrieval. We perform evaluations using the validation sets of Pouring, Penn Action, IKEA ASM and COIN. In particular, we consider each video from the validation set as a query video and treat all the remaining videos as a support set, following previous work [4]. For each frame in the query video, we retrieve its K most similar frames in the support set via nearest neighbor search in the embedding space. We report *Average Precision at* K, that is, the average percentage of the action label in the query frame, among all the actions in the K retrieved frames.

As depicted by Table 1, our approach, **VAVA**, consistently outperforms previous approaches for different values of K on all the evaluated datasets. These results demonstrate the effectiveness of our approach in learning fine-grained features and capturing the details of human actions.

Dataset	Model	AP@5	AP@10	AP@15
Pouring	SAL [5]	84.05	83.77	83.79
	TCN [6]	83.56	83.31	83.01
	TCC [2]	87.16	86.68	86.54
	LAV [4]	89.13	89.13	89.22
	VAVA(ours)	**90.05**	**89.92**	**90.17**
Penn Action	SAL [5]	76.04	75.77	75.61
	TCN [6]	77.84	77.51	77.28
	TCC [2]	76.74	76.27	75.88
	LAV [4]	79.13	78.98	78.90
	VAVA(ours)	**81.52**	**80.47**	**80.67**
IKEA ASM No Background	SAL [5]	15.15	14.90	14.72
	TCN [6]	19.15	19.19	19.33
	TCC [2]	19.80	19.64	19.68
	LAV [4]	23.89	23.65	23.56
	VAVA(ours)	**29.58**	**28.74**	**28.48**
IKEA ASM Background	SAL [5]	14.28	14.04	14.10
	TCN [6]	17.37	17.03	16.96
	TCC [2]	18.03	17.53	17.20
	LAV [4]	20.14	19.35	19.21
	VAVA(ours)	**26.42**	**25.73**	**25.80**
COIN	SAL [5]	26.83	25.95	25.84
	TCN [6]	27.05	26.92	26.59
	TCC [2]	28.58	28.05	28.34
	LAV [4]	29.39	28.48	28.20
	VAVA(ours)	**36.53**	**34.71**	**34.63**

Table 1. **Quantitative results for fine-grained frame retrieval.**

Dataset	Model	Classification	Progress	τ
Pouring	SAL [5]	85.86	0.6422	0.7329
	TCN [6]	85.98	0.6732	0.7500
	TCC [2]	88.59	0.7104	0.7774
	LAV [4]	87.70	0.7320	0.7867
	VAVA(ours)	**88.94**	**0.7627**	**0.8003**
Penn Action	SAL [5]	64.05	0.2989	0.4145
	TCN [6]	60.17	0.1909	0.4260
	TCC [2]	65.53	0.4304	0.4529
	LAV [4]	67.90	0.3853	0.4929
	VAVA(ours)	**69.71**	**0.4572**	**0.5291**
IKEA ASM No Background	SAL [5]	20.42	-	-
	TCN [6]	20.45	-	-
	TCC [2]	22.04	-	-
	LAV [4]	23.84	-	-
	VAVA(ours)	**26.52**	-	-
IKEA ASM Background	SAL [5]	20.83	-	-
	TCN [6]	21.70	-	-
	TCC [2]	21.43	-	-
	LAV [4]	22.05	-	-
	VAVA(ours)	**26.32**	-	-
COIN	SAL [5]	33.58	-	-
	TCN [6]	34.61	-	-
	TCC [2]	34.93	-	-
	LAV [4]	35.21	-	-
	VAVA(ours)	**40.68**	-	-

Table 2. **Quantitative results for training-from-scratch on all the benchmarks.**

2. Training-from-Scratch

In our paper, we provide results of our approach using models initialized with pre-trained weights from ImageNet classification, just like the baseline models we compare with [2, 4]. Here, we also provide experiments with models learned from scratch using a smaller backbone, VGG-M [1]. We use the same experimental setup with earlier works [2, 4]. As can be seen in Table 2, VAVA consistently outperforms the state-of-the-art.

3. Penn Action with All Categories

On the Penn Action dataset, following TCC [2], we evaluate the accuracy separately for each activity and report average performance in our main paper. Following [3, 4], we also evaluate with a single model trained on all the activities on this dataset. Table 3 shows results for this experimental setup. Our model outperforms previous work in this joint all-activity setting, which demonstrates that our approach is able to reliably align multiple actions with a single model.

Model	Classification	Progress	τ
SAL [5]	68.15	0.3903	0.4744
TCN [6]	68.09	0.3834	0.5417
TCC [2]	74.39	0.5914	0.6408
LAV [4]	78.68	0.6252	0.6835
GTA [3]	78.90	-	0.7484
VAVA(ours)	**80.25**	**0.6482**	**0.7620**

Table 3. **Joint all-action model results on Penn Action.**

Activity	Actions
Clean Hamster Cage	Remove the hamster from the hamster cage Remove the toy and paper bed from the hamster cage Clean toys and hamster cages Move the toy and paper bed into the hamster cage Put the hamster back into the hamster cage
Make RJ45 Cable	Strip the insulation Arrange the separated wire Cut a certain length Insert it into the crystal head Fix it with a crimping pliers
Change Mobile Phone Battery	Heat the back cover of the phone Pick up the back cover of the phone with the cymbal Remove the components of the fixed battery Remove the tape of the fixed battery Take down the old battery Put on new tape Load a new battery Restore the fixed battery components and the back cover
Attend NBA Skills Challenge	Do the first layup Dribble in the field Pass the basketball into the hole at the first time Shoot towards the basket Pass the basketball into the hole at the second time Dribble and lay up
Make Paper Wind Mill	Fold the edges of the paper Cut along the edges Fold the squares inward and fix them Fix the wind mill on the bracket
Replace Hard Disk	Open the laptop rear cover Remove the old hard disk Install the new hard disk Install the laptop rear cover

Table 4. **Summary of the major activities and actions for the evaluation on the COIN dataset.**

4. Performance per Activity on COIN

COIN [7] is a large-scale dataset and exhibits large temporal variations involving background frames, redundant frames and non-monotonic frames. This challenging dataset has not been used before by earlier work [2, 4] for the alignment task. On this dataset, we randomly select 6 major activities and report average performance in the main paper. The details of the selected major activities are summarized in Table 4. For each activity, we randomly select 60% of the sequences for training and the rest for evaluation. We present evaluation results for each activity in Table 5. Our approach, **VAVA**, outperforms the state-of-the-art methods on all the activities. The large margin in our improvement demonstrates the power of our approach in modeling temporal variations, which is particularly pronounced on COIN.

Activity	Model	Fraction of Labels 0.1	0.5	1.0
Clean Hamster Cage	Supervised Learning	35.04	38.28	40.54
	Random Features	30.03	31.11	31.04
	Imagenet Features	31.02	33.64	36.14
	SAL [5]	35.13	38.68	39.18
	TCN [6]	36.23	38.93	40.19
	TCC [2]	35.82	38.64	40.11
	LAV [4]	32.25	31.94	32.32
	VAVA(ours)	**43.12**	**40.74**	**43.39**
Make RJ45 Cable	Supervised Learning	36.74	38.22	49.44
	Random Features	30.45	32.01	31.19
	Imagenet Features	32.09	34.14	39.02
	SAL [5]	36.25	40.6	42.71
	TCN [6]	37.49	41.93	42.72
	TCC [2]	37.04	40.27	42.51
	LAV [4]	38.52	41.98	45.41
	VAVA(ours)	**45.29**	**46.16**	**47.6**
Change Mobile Phone Battery	Supervised Learning	31.91	36.02	39.31
	Random Features	30.01	30.26	30.95
	Imagenet Features	31.03	31.59	33.17
	SAL [5]	34.33	37.59	39.10
	TCN [6]	34.35	37.72	38.25
	TCC [2]	34.02	36.48	38.17
	LAV [4]	32.07	31.64	34.15
	VAVA(ours)	**37.16**	**38.47**	**39.13**
Attend NBA Skills Challenge	Supervised Learning	49.9	56.69	73.85
	Random Features	26.01	27.05	27.13
	Imagenet Features	33.18	39.17	40.24
	SAL [5]	34.35	40.58	41.97
	TCN [6]	37.32	40.71	42.3
	TCC [2]	38.96	42.42	43.73
	LAV [4]	43.63	44.64	45.62
	VAVA(ours)	**54.96**	**64.59**	**66.29**
Make Paper Wind Mill	Supervised Learning	33.07	36.75	52.01
	Random Features	30.41	30.95	31.03
	Imagenet Features	30.32	34.41	39.02
	SAL [5]	35.48	40.29	41.38
	TCN [6]	31.92	40.85	42.58
	TCC [2]	34.13	41.25	42.27
	LAV [4]	38.58	44.58	44.5
	VAVA(ours)	**45.07**	**47.81**	**49.01**
Replace Hard Disk	Supervised Learning	35.99	38.44	39.92
	Random Features	30.01	30.35	30.94
	Imagenet Features	30.27	35.47	36.97
	SAL [5]	32.60	37.66	37.60
	TCN [6]	31.91	38.24	37.03
	TCC [2]	35.26	38.31	37.17
	LAV [4]	35.69	38.33	36.84
	VAVA(ours)	**37.04**	**39.33**	**38.14**

Table 5. **Activity-wise evaluation on the COIN dataset.**

5. Additional Ablation Studies and Discussion

Hyperparameters γ, λ_1 and λ_2. In this section, we provide additional ablation studies regarding the hyper-parameters we used in our model, namely, γ in Eq. 17, λ_1 and λ_2 in Eq. 13. Following the ablation studies in our main paper (Table 2), we evaluate on the IKEA ASM dataset with background frames. As shown by Table 6, $\gamma = 0.5$, $\lambda_1 = 1.0$ and $\lambda_2 = 0.1$ yield the best performance on average, we therefore use this setting for the experiments in our paper. The results also suggest that our approach produces similar accuracies for different sets of γ, λ_1 and λ_2, and, hence, is not sensitive against the choice of hyper-parameters.

Hyperparameter	Value	Fraction of Labels 0.1	0.5	1.0
γ	0.1	27.73	28.05	28.42
	0.5	**29.12**	**29.95**	**29.10**
	1.0	28.20	28.53	28.79
λ_1	0.5	28.32	29.03	**29.43**
	1.0	**29.12**	**29.95**	29.10
	1.5	28.65	28.94	29.31
λ_2	0.01	28.08	29.05	29.17
	0.1	**29.12**	**29.95**	29.10
	1.0	28.07	29.12	**29.29**

Table 6. **Ablation for hyper-parameters using IKEA ASM.**

Backbone. In this paper, we use ResNet-50 to follow the exact same setup as previous work for a fair comparison. To demonstrate that our approach is not limited to a specific backbone, we ablate with the R3D-18 [8] backbone and form a single block of 5 consecutive frames, accounting for the maximum amount of the memory of our GPU. We then learn to align the middle frame of the block for frame-wise alignment on the IKEA dataset. As shown in Table 7(a), **VAVA** still outperforms previous work with this new backbone and the 3D backbone consistently improves the performance as it provides more contextual information.

Distribution. Instead of GMM, we also experimented with a flat-top Gaussian distribution as in Eq. 1, where c is a scale value for normalization and η is a margin value in which we can allow for temporal variations. We also evaluate with Beta distribution as in Eq. 2. We ablate with different hyper-parameters and report the best performances in Table 7(c). GMM outperforms them and we therefore use it in our experiments.

$$f(x) = \begin{cases} \frac{c}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}, & \text{if } |x-\mu| > \eta \\ \frac{c}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{\eta}{\sigma})^2}, & \text{o.w.} \end{cases} \quad (1)$$

$$g(x) = c \cdot x^{\alpha-1}(1-x)^{\beta-1} \quad (2)$$

Variance. We also conduct an ablation study on the effect of different variances. In addition, we build a simple extension of our model with extra MLP layers on video features that learn the corresponding variance of the distribution. We report our results on the IKEA dataset in Table 7(b). As can be seen by the analysis, a larger variance does not necessarily bring better performance as it may degrade the impact of temporal priors and the learned variance perform slightly worse than the fine-tuned fixed value as learning the variance itself is not an easy task, especially without any frame-wise labels.

Model	Fraction of Labels 0.1	0.5	1.0
SAL	23.61	24.73	26.91
TCN	23.20	24. 93	26.41
TCC	23.84	25.61	27.03
LAV	24.27	26.33	26.61
VAVA	**31.76**	**31.88**	**32.07**

(a)

Variance	Fraction of Labels 0.1	0.5	1.0
0.5^2	28.63	28.91	29.05
0.6^2	**29.12**	**29.95**	**29.10**
0.7^2	28.49	29.02	28.72
Learned	28.71	29.25	28.84

(b)

Distribution	Fraction of Labels 0.1	0.5	1.0
flat-top	27.59	28.02	28.24
Beta	27.66	28.18	27.93
GMM	**29.12**	**29.95**	**29.10**

(c)

Table 7. Ablation on (a) backbone, (b) variance and (c) distribution.

Robustness when aligning videos with different durations. In our approach, we randomly sample equal number of frame indices and use the sorted indices to extract frames from two videos. This sampling strategy allows us to train our model with pairs of videos that have different lengths and brings in robustness to changes in the speeds of actions and temporal variations across sequences. This consequently allows us to reliably align two sequences even when there is a large difference in their lengths, such as those of the COIN dataset, on which the sequence length varies to a large extent, i.e. from 29 to 527 seconds.

Hyperparameter	Value
Batch Size	4
Number of frames	40(C,P), 20(PA,IA)
Optimizer	ADAM
Learning Rate	1.0×10^{-4}
Weight Decay	1.0×10^{-5}
Window Size(δ)	15
γ	0.5
λ_1	1.0
λ_2	0.1

Table 8. **List of hyperparameters.** C, P, PA and IA represent COIN, Pouring, Penn Action and IKEA ASM, respectively.

6. Implementation Details

We follow previous work [2, 4] and use the same backbone network for a fair comparison. We provide a list of values for our hyper-parameters in Table 8. and summarize our network architecture in Table 9.

7. Visualization

t-SNE Visualization. We present an example of t-SNE [9] visualization of the embeddings in Fig. 1. Frames with the same border color are sampled from different time-steps in

Model	Layer	Output Size	Parameter
Base Network	conv1	$112 \times 112 \times 64$	$7 \times 7, 64$, stride 2 3×3 max pool, stride 2
	conv2	$56 \times 56 \times 256$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
	conv3	$28 \times 28 \times 512$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
	conv4	$14 \times 14 \times 1024$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 3$
Embedder Network	Temporal Stacking	$1 \times 14 \times 14 \times 1024$	Stack 1 context frame features in time axis
	conv5	$1 \times 14 \times 14 \times 512$	$\begin{bmatrix} 3 \times 3 \times 3, 512 \\ 3 \times 3 \times 3, 512 \end{bmatrix} \times 1$
	Spatio-temporal Pooling	512	Global 3D Max-Pool
	fc6	512	$\begin{bmatrix} 512 \\ 512 \end{bmatrix} \times 1$
	Embedding	128	128

Table 9. **Model architecture in our experiments.** The network produces an embedding for each frame. We show different network parameters inside the square brackets using the following formalism: (1) $[n \times n, c]$ refers to 2D Convolution filter size of n and the number of channels of c; (2)$[n \times n \times n, c]$ refers to 3D Convolution filter size of n and the number of channels of c; (3) $[c]$ refers to c channels in a fully-connected layer.

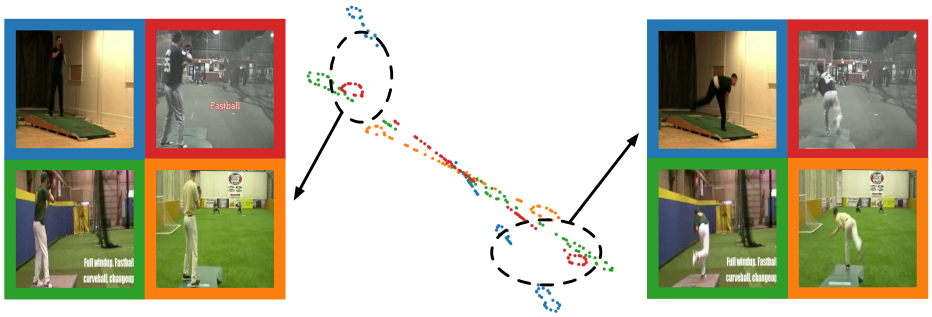

Figure 1. **Embeddings.** We visualize the embeddings with t-SNE [9] using videos from the 'baseball pitch' action of Penn Action.

the same video. The visualization depicts how the embeddings change as an action is being carried out. As depicted by Fig. 1, our approach not only can align actions from different videos but also can capture the appearance and motion variations, which is crucial for fine-grained understanding of actions.

Alignment. We further provide an accompanying demo video ("demo.mp4") which shows the capability of our approach, **VAVA**, in reliably aligning two complex video sequences on all the benchmarks. Even though there are heavy temporal variations and appearance changes across sequences, our approach is still able to align them without using any frame-wise labels.

8. Discussion

Limitations and Future Work. Previous work on representation learning by sequence alignment [2,4] requires two videos with the same major activity using a weak supervision setting. To have a fair a comparison against them, we also follow the same setup. While this is a limitation of the existing approaches and our method, one potential solution to this problem is to leverage a pre-trained action recognition model for determining if the two videos contain the same major activity, and, only then, to use alignment across those two sequences for fine-grained action understanding.

Although we significantly outperform earlier approaches in downstream tasks on all the datasets, the accuracy numbers on the IKEA ASM and COIN datasets suggest that further improvement is required for practical deployment of this technology. One straightforward way to achieve higher accuracy could be to use a stronger backbone network that captures temporal contextual information. While we use a ResNet-50 backbone to have a fair comparison against [2, 4], future work will examine the influence of different backbone networks on the accuracy of sequence alignment and downstream tasks.

Societal Impact. Sequence alignment and fine-grained action understanding are important for applications in procedure learning and robot imitation learning, which has many beneficial use cases in AR-based task guidance, assistive technologies for handicapped people and industrial automation. However, it can also be used for monitoring people's daily activities and cause privacy issues, therefore it requires mindful deployment of technology.

References

[1] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *arXiv Preprint*, 2014. 2

[2] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal Cycle-Consistency Learning. In *Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 3, 4

[3] Isma Hadji, Konstantinos G. Derpanis, and Allan D. Jepson. Representation Learning via Global Temporal Alignment and Cycle-Consistency. In *Conference on Computer Vision and Pattern Recognition*, 2021. 2

[4] Sanjay Haresh, Sateesh Kumar, Huseyin Coskun, Shahram Najam Syed, Andrey Konin, Muhammad Zeeshan Zia, and Quoc-Huy Tran. Learning by Aligning Videos in Time. In *Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2, 3, 4

[5] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Shuffle and Learn: Unsupervised Learning Using Temporal Order Verification. In *European Conference on Computer Vision*, 2016. 1, 2

[6] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-Contrastive Networks: Self-Supervised Learning from Video. In *International Conference on Robotics and Automation*, 2018. 1, 2

[7] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. COIN: A Large-scale Dataset for Comprehensive Instructional Video Analysis . In *Conference on Computer Vision and Pattern Recognition*, 2019. 2

[8] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *Conference on Computer Vision and Pattern Recognition*, 2018. 3

[9] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data Using t-SNE. *Journal of Machine Learning Research*, 2008. 3, 4