Supplementary Material: Mitigating and Evaluating Static Bias of Action Representations in the Background and the Foreground

Haoxin Li[1], Yuan Liu[2], Hanwang Zhang[1], Boyang Li[1]
[1]Nanyang Technological University [2]Guangzhou University
{haoxin003, hanwangzhang, boyang.li}@ntu.edu.sg, yuanliu@gzhu.edu.cn

Contents

Figure S1: Examples of extracted foreground images.

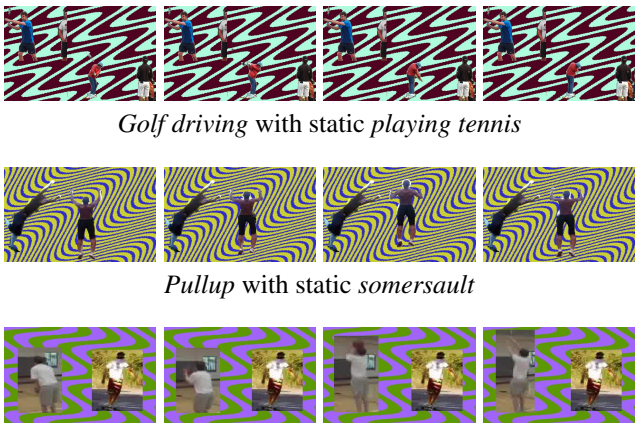


Figure S2: Examples of the videos with conflicting foreground cues.

S1. Experimental Results and Analysis

S1.1. Extracted Foreground Images

In Figure S1, we show several extracted foreground images using the foreground masks in Kinetics-400. From these images, we observe that the foreground motion and the foreground static cues share the same pixels. Therefore, unlike directly separating the pixels of foregrounds and backgrounds in ActorCutMix [24] and FAME [5], it

Table S1: Action recognition accuracy (%) of different augmentation and debiasing methods on videos with conflicting foreground cues.

Augmentation or Debiasing	Kinetics-400		UCF101		HMDB51	
	Multi-class	Binary	Multi-class	Binary	Multi-class	Binary
No	25.25	72.83	44.65	81.94	36.58	85.03
Mixup	27.64	74.48	47.16	81.85	36.62	82.06
VideoMix	29.37	72.50	42.59	72.21	32.68	76.07
SDN	27.14	71.14	48.47	83.83	34.87	81.88
BE	26.67	72.99	46.62	81.73	35.99	85.30
ActorCutMix	29.02	74.02	56.88	79.60	36.97	81.07
FAME	29.50	73.83	28.21	71.70	39.61	81.56
StillMix	**30.77**	**85.51**	**57.30**	**88.80**	**47.38**	**92.46**

Table S2: Action recognition accuracy (%) of different augmentation and debiasing methods on ARAS.

Augmentation or Debiasing	Main Network		
	TSM	SlowFast	Swin-T
No	57.86	50.14	60.17
Mixup	58.05	50.63	59.59
VideoMix	56.61	47.44	60.95
SDN	55.06	48.80	60.26
BE	57.47	50.92	59.79
ActorCutMix	57.09	**51.40**	61.23
FAME	57.47	48.51	60.37
StillMix (Ours)	**59.69**	**51.40**	**62.49**

is difficult to separate the pixels of foreground motion and foreground static cues for debiasing foreground static bias. As a result, in this paper, we propose StillMix to debias without the need to explicitly extract foreground static cues within a frame. In addition, due to this difficulty, it is hard to create test videos by simply replacing the foreground static cues and preserving the foreground motion. Thus, we alternatively create videos with conflicting foreground cues (Figure 1 of the main paper) and SCUFO videos (Sec. 4 of the main paper) to evaluate foreground static bias.

S1.2. Testing on Videos with Conflicting Foreground Cues

A video with conflicting foreground cues is synthesized from a SCUBA-Sinusoid video by the following steps:

1. Randomly sample a video with foreground masks but different action label from the SCUBA-Sinusoid video.
2. Randomly sample a frame in the sampled video and use the foreground mask to extract the foreground (mainly containing human actors) as a static foreground.
3. Randomly select a spatial position in the SCUBA-Sinusoid video to insert the static foreground such that the inserted static foreground does not overlap with the moving foreground.
4. Insert the static foreground into all the frames of the SCUBA-Sinusoid video at the selected spatial position.
5. Resize the resultant video to the size of the SCUBA-Sinusoid video.
6. Keep the label of the resultant video as the same as the SCUBA-Sinusoid video.

The resultant video contains two action features, one on the static foreground and the other on the moving foreground. We show some example videos in Figure S2. A robust action recognition model should not be affected by the inserted static foregrounds and obtain high accuracy.

We use two metrics to evaluate the performance on the videos with conflicting foreground cues: (1) Multi-class classification accuracy: each video is classified into N action classes, where N is the number of classes defined in the datasets. (2) Binary classification accuracy: each video is classified into two action classes, one indicating the action in the moving foreground and the other indicating the "action" in the static foreground.

Table S1 shows the accuracies of different data augmentation and debiasing methods with Swin-T as the base model. From the results, we observe that StillMix obtains the best performance, especially in binary classification (*i.e.*, outperforming other methods by more than 4%). Although ActorCutMix and FAME outperform StillMix on SCUBA videos (refer to Table 2 of the main paper and Table S10 in the Supplementary Material), they perform worse than StillMix on the videos with conflicting foreground cues. The results indicate that FAME and ActorCutMix capture foreground static features as shortcuts instead of learning robust motion features; when the static foregrounds exist, they are interfered to predict the static "action". In contrast, StillMix shows better robustness to the foreground static features.

S1.3. Testing on ARAS

To assess the effectiveness of different methods on mitigating scene bias, we conduct tests on a real-world OOD video dataset, ARAS [6], which contains actions defined in Kinetics-400 with rare scenes.

After trained on Kinetics-400, the models are directly tested on a balanced test set of ARAS as in [6]. Table S2 shows the accuracies of different data augmentation and debiasing methods. From the results, we observe that StillMix obtains the best performance, illustrating its effectiveness on mitigating scene bias in real-world videos.

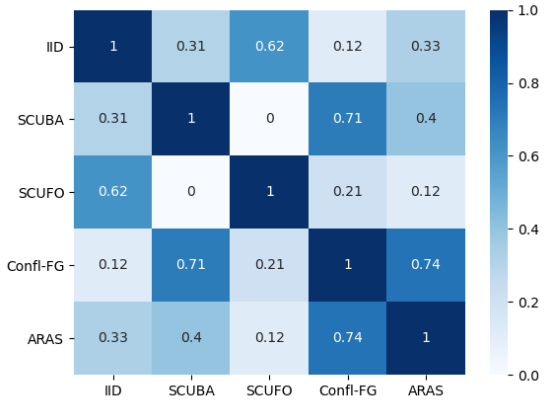


Figure S3: Correlations between different evaluations are not strong. Confl-FG denotes videos with conflicting foreground cues.

Table S3: IID and OOD test accuracy (%) of StillMix based on UniformerV2.

Dataset	Debiasing	IID	OOD				
			Avg SCUBA ↑	Avg SCUFO ↓	Contra. Acc. ↑	Confl-FG ↑	ARAS ↑
Kinetics-400	No	88.12	69.21	44.44	27.06	42.11	81.77
	StillMix	**88.32**	**70.42**	**40.90**	**31.59**	**42.96**	**83.12**
HMDB51	No	**82.94**	61.73	50.75	15.43	34.10	–
	StillMix	82.35	**62.21**	**49.75**	**16.59**	**34.49**	–
UCF101	No	98.18	63.65	38.73	26.45	47.58	–
	StillMix	**98.36**	**64.56**	**38.61**	**27.53**	**48.07**	–

S1.4. Correlations between Different Evaluations

To determine the level of similarity between different evaluations, we calculate the Spearman's rank correlation coefficients between the performances of different methods on any pair of evaluation data. We use the performance of Swin-T on Kinetics-400 for calculation. The results are shown in Figure S3, from which we observe that the correlation coefficients are lower than 0.75, indicating that the correlations between any pair of the evaluation data are not strong. Therefore, these evaluation data may assess the action representations from different perspectives and could not directly replace each other.

S1.5. Performance of UniformerV2

With StillMix, we finetune UniformerV2 [12], the SOTA opensource action recognition model. We finetune UniFormerV2-L/14 from CLIP and K710 pretrained weights, using 32 frames as inputs. The results are shown in Table S3. As UniformerV2 adopts pretrained weights from CLIP, an image-only network, it has strong static bias and performs poorly on SCUFO and Confl-FG. For example, Swin-T achieves 36.58% on HMDB51-Confl-FG (Table S1) but UniformerV2, having gone through much more pretraining, achieves only 34.10%. Hence, finetuning with StillMix on small datasets like HMDB51 and UCF101 could not substantially correct this bias, but StillMix still shows improvements over the original model.

S1.6. Full Results of Transfer Learning

To evaluate the robustness of the learned action representations, we conduct tests of transferring action features across datasets. The rationale for this evaluation is that the static bias is likely idiosyncratic to the dataset and may not transfer well across datasets or class definitions. In comparison, the motion features should transfer well across datasets and class definitions. We adopt the linear probing protocol. After training on the source dataset, we fix the backbone network and train only a linear classifier on top of the backbone using the target dataset.

Table S4 shows the performance of different data augmentation and debiasing methods with different base models. From the results, we observe that the models trained with StillMix obtain the best performance in different transferring settings, especially in transferring across small datasets. For example, in transferring from HMDB51 to UCF101, StillMix outperforms other data augmentation methods by about 2% of accuracy. These results illustrate that StillMix learns robust action representations that have better capability to transfer across action datasets.

S1.7. Ablation Study

In this section, we provide more results of ablation studies.

Sampling biased frames improves debiasing. In Table S5, we show more results of different frame sampling strategies on HMDB51 using different main networks and pretraining datasets. As in the main paper, we compare three frame sampling strategies: (1) *No RefNet*; (2) *RefNet*; (3) *RefNet Inversed*. Comparing the results of *RefNet Inversed* with the other two strategies, we observe that *RefNet Inversed* obtains significantly lower OOD performance especially for ImageNet pretrained models (more than 3%). Comparing the results of *No RefNet* and *RefNet*, we observe that they obtain similar IID performance but *RefNet* performs better on OOD tests especially for ImageNet pretrained TSM and Swin-T (more than 2%). The results show that sampling biased frames benefits bias mitigation. Even sampling frames with *RefNet Inversed* or *No RefNet*, StillMix still outperforms other methods (refer to the results in Table S9), further indicating its effectiveness on bias mitigation.

Mixing action labels in StillMix decreases performance.

Table S4: Action recognition accuracy (%) of transferring the learned representations across datasets.

Network	Augmentation or Debiasing	Source→Target			
		Kinetics400→UCF101	Kinetics400→HMDB51	HMDB51→UCF101	UCF101→HMDB51
TSM	No	92.52	66.67	61.64	44.95
	Mixup	93.07	68.69	63.58	46.60
	VideoMix	93.55	69.22	61.49	40.33
	SDN	92.81	63.79	61.12	41.90
	BE	93.10	67.45	62.71	45.88
	ActorCutMix	92.73	67.39	61.67	42.92
	FAME	93.87	67.84	58.87	44.99
	StillMix	**93.89**	**70.07**	**65.69**	**47.99**
SlowFast	No	91.86	67.32	42.38	40.59
	Mixup	90.14	65.49	42.63	43.86
	VideoMix	89.80	64.25	42.90	39.30
	SDN	89.29	61.30	43.46	38.91
	BE	91.91	67.12	40.89	41.29
	ActorCutMix	91.73	67.19	43.66	39.17
	FAME	91.01	65.10	39.49	39.37
	StillMix	**92.49**	**67.84**	**46.23**	**44.77**
Swin-T	No	95.74	72.03	75.67	52.83
	Mixup	95.40	72.42	76.61	51.59
	VideoMix	95.32	71.24	74.95	50.59
	SDN	94.90	70.13	74.42	49.87
	BE	95.43	71.63	76.78	53.31
	ActorCutMix	95.72	72.55	75.57	52.81
	FAME	95.40	70.78	75.67	50.55
	StillMix	**95.77**	**72.75**	**78.50**	**53.71**

Table S5: Action recognition accuracy (%) of StillMix with different frame sampling strategies.

Network	Pretrain	Sampling strategy	HMDB51	
			IID	Contra. Acc.
TSM	ImageNet	No RefNet	54.07	31.21
		RefNet	54.66	**33.14**
		RefNet Inversed	54.79	29.17
	Kinetics400	No RefNet	71.87	41.86
		RefNet	71.52	**42.05**
		RefNet Inversed	72.27	38.98
SlowFast	ImageNet	No RefNet	51.79	20.94
		RefNet	51.53	**21.21**
		RefNet Inversed	50.94	18.61
	Kinetics400	No RefNet	76.27	34.34
		RefNet	76.52	**35.20**
		RefNet Inversed	76.12	33.83
Swin-T	ImageNet	No RefNet	55.62	18.89
		RefNet	55.36	**21.40**
		RefNet Inversed	56.34	18.18
	Kinetics400	No RefNet	75.16	39.66
		RefNet	74.82	**40.28**
		RefNet Inversed	75.62	37.82

Table S6: Action recognition accuracy (%) of StillMix with label mixing. $\lambda' = 1$ means StillMix without label mixing (the default setting).

Pretrain	λ'	UCF101		HMDB51	
		IID	Contra. Acc.	IID	Contra. Acc.
ImageNet	1	87.29	**24.60**	54.66	**33.14**
	0.8	86.40	14.56	54.44	22.69
	λ	84.35	10.70	49.76	5.06
Kinetics400	1	94.30	**36.47**	71.52	**42.05**
	0.8	94.70	31.54	72.07	35.19
	λ	93.85	18.82	70.92	19.44

Table S7: Action recognition accuracy (%) of StillMix with different Beta distributions parameters. IN and K400 denote ImageNet and Kinetics-400 respectively.

Pretrain	$Beta(\alpha, \beta)$	Mean of λ	Variance of λ	UCF101		HMDB51	
				IID	Contra. Acc.	IID	Contra. Acc.
IN	$(300, 100)$	0.75	0.00047	85.98	12.43	50.70	7.95
	$(100, 300)$	0.25	0.00047	87.24	22.05	54.92	**38.06**
	$(200, 200)$	0.5	0.00062	87.29	**24.60**	54.66	33.14
	$(100, 100)$	0.5	0.0012	87.41	22.20	55.08	32.18
	$(20, 20)$	0.5	0.0060	87.42	23.01	54.95	33.06
K400	$(300, 100)$	0.75	0.00047	94.49	27.58	71.65	26.22
	$(100, 300)$	0.25	0.00047	94.36	**37.38**	72.29	**49.41**
	$(200, 200)$	0.5	0.00062	94.30	36.47	71.52	42.05
	$(100, 100)$	0.5	0.0012	94.41	36.64	72.18	41.94
	$(20, 20)$	0.5	0.0060	94.60	36.12	71.85	42.69

Table S8: Action recognition accuracy (%) of different augmentation and debiasing methods on the Kinetics-400, Kinetics400-SCUBA and Kinetics400-SCUFO datasets. † indicates adapting from self-supervised debiasing methods.

Model	Pretrain	Augmentation or Debiasing	Kinetics-400	Kinetics400-SCUBA (↑)			Kinetics400-SCUFO (↓)			Contra. Acc. (↑)
				Place365	VQGAN-CLIP	Sinusoid	Place365	VQGAN-CLIP	Sinusoid	
TSM	ImageNet	No	71.13	40.44	37.39	34.34	18.55	16.61	16.49	22.80
		Mixup	71.33	42.82	40.65	38.95	18.66	17.20	16.72	25.98
		VideoMix	**71.35**	40.93	38.96	36.71	17.89	17.13	16.73	24.57
		SDN	69.99	38.42	35.71	36.72	16.98	15.31	17.35	22.38
		BE†	71.30	41.07	38.19	34.42	17.05	16.14	15.04	24.35
		ActorCutMix†	71.07	42.89	40.89	37.47	17.60	15.63	15.65	26.52
		FAME†	71.13	43.30	40.41	**39.01**	18.96	17.73	18.32	25.63
		StillMix (Ours)	71.28	**43.31**	**40.97**	37.15	**6.27**	**5.17**	**4.24**	**36.07**
SlowFast	ImageNet	No	65.63	36.96	35.54	34.88	21.74	20.19	20.51	18.98
		Mixup	65.16	37.65	35.63	34.98	20.85	18.81	18.81	20.17
		VideoMix	64.26	35.23	33.81	34.29	19.84	18.05	18.57	19.41
		SDN	63.49	33.71	31.60	31.86	19.81	18.21	19.69	17.13
		BE†	65.65	36.15	34.66	33.64	19.13	17.25	18.30	20.15
		ActorCutMix†	**65.79**	39.61	**38.28**	36.05	20.08	18.58	18.95	22.01
		FAME†	65.13	**40.11**	37.48	**38.63**	20.89	19.21	20.79	22.07
		StillMix (Ours)	65.65	37.63	35.71	35.46	**14.78**	**13.35**	**12.70**	**25.01**
Swin-T	ImageNet	No	73.95	40.81	39.67	44.75	17.89	15.89	20.73	25.93
		Mixup	73.91	42.59	42.05	47.22	17.33	15.74	20.68	28.24
		VideoMix	73.80	41.89	41.00	46.63	18.72	16.36	22.71	26.40
		SDN	72.23	39.59	39.90	47.52	20.04	19.21	25.13	24.46
		BE†	73.93	41.88	41.86	46.47	18.73	17.12	22.84	26.28
		ActorCutMix†	**73.97**	44.16	45.06	47.87	19.58	17.06	21.54	28.64
		FAME†	73.81	**49.01**	**47.71**	**49.66**	21.38	19.10	23.33	30.03
		StillMix (Ours)	73.86	43.44	42.81	46.05	**4.76**	**4.37**	**7.41**	**39.41**

StillMix keeps the label unchanged after augmentation. Here, we investigate the effects of mixing action labels, *i.e.*, $\tilde{y}_i = \lambda' y_i + (1-\lambda')y^{\text{biased}}$ where y^{biased} is the action label of the biased frame $\boldsymbol{z}^{\text{biased}}$. In Table S6, we compare the performance of different values of λ' on UCF101 and HMDB51 using TSM as the main network. We observe that mixing action labels significantly decreases the OOD performance although it could slightly boost the IID performance for Kinetics400 pretrained TSM by around 0.5% of accuracy. The results illustrate that mixing action labels in StillMix is detrimental to learning robust action representations, since it encourages models to learn biased static cues that are not robust in OOD scenarios.

Effects of Beta Distribution in StillMix. With different Beta distribution parameters in StillMix, the mixing coefficient λ (in Eq. (3) of the main paper) has different values of mean and variance. In Table S7, we compare the performance of different Beta distribution parameters on UCF101 and HMDB51 using ImageNet pretrained TSM as the main network. Comparing the results of different mean values of λ, we observe that both IID and OOD performance decrease when the mean value is large (*e.g.*, 0.75). With large values of λ, the mixed videos approximate the original videos, so that the debiasing effects are weak. In contrast, small mean values of λ (*e.g.*, 0.5, 0.25) lead to good IID and OOD performance. The results indicate that sufficient mixing strength is necessary for StillMix to mitigate static bias. Comparing the results of different variances of λ, we observe that increasing the variances improves the performance. The reason may be that large variances could augment videos with various mixing strength, which creates diverse augmented samples that help training.

S1.8. Full Results of Debiasing Methods

Table S8, S9 and S10 show the full results of different video data augmentation and debiasing methods on SCUBA and SCUFO videos of the Kinetics-400, HMDB51 and UCF101 datasets, respectively. The observations are similar to that in the main paper.

S1.9. Evaluation of Pretraining Methods

In this section, we evaluate several pretraining methods on the synthetic OOD data to demonstrate how pretraining affects OOD generalization. We evaluate the following pretraining methods:

Debiasing Pretraining Methods: (1) SDN [2], a supervised debiasing pretraining method that minimizes scene information and maximizes human action information using adversarial classifiers. (2) FAME [5], a self-supervised debiasing pretraining method which carves out the foreground

Table S9: Action recognition accuracy (%) of different augmentation and debiasing methods on the HMDB51, HMDB51-SCUBA and HMDB51-SCUFO datasets. K400 denotes Kinetics-400. † indicates adapting from self-supervised debiasing methods.

Model	Pretrain	Augmentation or Debiasing	HMDB51	HMDB51-SCUBA (↑)			HMDB51-SCUFO (↓)			Contra. Acc. (↑)
				Place365	VQGAN-CLIP	Sinusoid	Place365	VQGAN-CLIP	Sinusoid	
TSM	K400	No	70.39	45.09	42.16	26.84	23.26	20.03	14.40	22.02
		Mixup	**72.00**	46.25	44.07	28.96	22.60	19.92	14.71	23.76
		VideoMix	70.72	42.68	41.46	23.00	20.98	18.99	12.46	21.03
		SDN	69.51	40.79	38.92	31.44	19.18	14.91	18.70	23.74
		BE†	71.22	45.39	42.81	27.25	23.42	20.52	14.40	22.39
		ActorCutMix†	70.52	45.81	42.32	27.08	23.29	20.43	15.12	21.94
		FAME†	70.39	52.03	**53.21**	36.33	26.04	23.34	17.60	28.21
		StillMix (Ours)	71.52	**53.91**	52.66	**38.13**	**11.63**	**8.29**	**5.38**	**42.05**
SlowFast	K400	No	76.25	47.36	48.23	35.00	23.26	23.14	18.78	28.37
		Mixup	75.69	48.72	50.20	35.38	24.42	23.99	19.60	28.22
		VideoMix	75.62	48.19	48.87	34.81	23.98	23.80	19.17	27.25
		SDN	76.17	34.27	37.80	30.41	**9.39**	**12.04**	**12.30**	24.99
		BE†	75.82	46.46	47.29	34.51	23.50	23.46	19.20	27.74
		ActorCutMix†	75.49	53.28	52.94	38.96	26.57	26.42	21.28	28.96
		FAME†	74.66	**57.61**	**58.27**	**47.08**	25.68	24.02	21.52	34.32
		StillMix (Ours)	**76.52**	48.31	47.75	40.31	17.76	17.51	15.65	**35.20**
Swin-T	K400	No	73.92	47.61	42.77	41.41	20.68	17.90	22.80	27.84
		Mixup	74.58	46.70	42.49	40.12	21.25	18.47	23.78	26.09
		VideoMix	73.31	41.33	38.18	38.67	19.64	18.82	22.85	23.13
		SDN	74.66	41.96	40.82	37.29	19.99	19.62	21.06	22.88
		BE†	74.31	47.36	42.94	40.39	20.91	17.55	21.41	27.84
		ActorCutMix†	74.05	50.13	46.51	43.73	22.16	20.26	23.80	28.12
		FAME†	73.79	**54.71**	**53.67**	45.81	27.10	27.26	26.40	29.66
		StillMix (Ours)	**74.82**	53.27	52.43	**49.73**	**13.39**	**12.66**	**14.13**	**40.28**
TSM	ImageNet	No	47.56	19.39	16.99	8.49	11.78	11.12	6.41	6.50
		Mixup	51.68	23.96	18.76	14.06	18.49	15.04	12.10	5.66
		VideoMix	48.32	21.41	18.44	10.27	13.23	12.42	7.64	7.17
		SDN	45.40	22.57	16.63	13.11	11.37	8.08	8.15	10.72
		BE†	48.87	22.13	16.86	13.11	17.39	14.33	11.39	4.66
		ActorCutMix†	48.39	25.52	21.38	11.57	15.27	13.16	8.39	9.03
		FAME†	45.73	26.37	24.34	15.46	14.69	15.69	10.44	10.71
		StillMix (Ours)	**54.66**	**39.52**	**38.41**	**33.00**	**6.98**	**6.09**	**3.58**	**33.14**
SlowFast	ImageNet	No	47.65	21.24	16.85	10.43	17.84	15.63	11.76	5.62
		Mixup	48.67	23.22	18.75	12.60	19.57	16.19	12.41	5.70
		VideoMix	47.38	21.11	17.68	9.53	16.78	15.57	10.92	5.49
		SDN	44.29	18.05	13.07	11.49	16.31	**12.07**	11.21	3.06
		BE†	45.10	19.40	17.21	8.54	**13.77**	14.35	**8.52**	7.73
		ActorCutMix†	49.21	28.40	25.16	15.67	21.64	20.30	15.53	8.34
		FAME†	45.97	27.83	27.29	17.12	20.26	20.86	14.87	10.18
		StillMix (Ours)	**51.53**	**33.55**	**32.14**	**25.81**	14.34	13.68	12.59	**21.21**
Swin-T	ImageNet	No	53.62	23.45	19.39	18.24	18.25	14.80	15.94	6.56
		Mixup	55.86	25.42	18.96	17.12	20.53	14.50	15.22	6.66
		VideoMix	**56.17**	26.31	21.98	18.02	19.60	17.24	16.14	7.89
		SDN	53.16	22.29	18.28	17.59	19.71	15.98	15.69	4.96
		BE†	53.90	23.32	17.64	14.49	18.74	13.88	13.46	5.51
		ActorCutMix†	54.07	29.23	25.16	22.59	22.36	19.08	17.75	8.84
		FAME†	53.18	22.29	26.46	23.88	23.15	19.91	19.39	9.42
		StillMix (Ours)	55.36	**34.73**	**30.84**	**30.83**	**16.13**	**11.60**	**13.20**	**21.40**

Table S10: Action recognition accuracy (%) of different augmentation and debiasing methods on the UCF101, UCF101-SCUBA and UCF101-SCUFO datasets. K400 denotes Kinetics-400. † indicates adapting from self-supervised debiasing methods.

Model	Pretrain	Augmentation or Debiasing	UCF101	UCF101-SCUBA (↑)			UCF101-SCUFO (↓)			Contra. Acc. (↑)
				Place365	VQGAN-CLIP	Sinusoid	Place365	VQGAN-CLIP	Sinusoid	
TSM	K400	No	94.62	26.79	22.66	27.36	5.63	4.12	2.89	21.83
		Mixup	**94.71**	29.03	24.85	29.51	5.20	3.94	2.99	24.17
		VideoMix	94.50	32.76	29.99	31.89	6.73	5.63	4.94	26.69
		SDN	93.83	22.15	18.37	19.22	3.46	2.53	3.32	17.19
		BE†	94.49	27.25	22.99	27.52	6.07	4.42	3.36	21.82
		ActorCutMix†	94.47	**38.95**	37.63	37.74	4.84	4.85	3.99	33.90
		FAME†	93.73	36.80	**37.76**	32.61	4.63	4.10	2.28	32.28
		StillMix (Ours)	94.30	37.40	33.85	**40.30**	**0.97**	**0.81**	**0.60**	**36.47**
SlowFast	K400	No	95.96	34.34	31.00	30.19	2.44	1.51	1.14	30.25
		Mixup	**96.14**	36.60	33.20	32.58	4.50	2.89	2.81	30.94
		VideoMix	95.98	38.71	39.90	31.03	4.85	3.86	3.80	31.57
		SDN	95.02	32.24	29.25	24.32	4.64	3.02	1.95	25.72
		BE†	95.98	35.24	31.31	30.66	3.02	2.09	1.72	30.24
		ActorCutMix†	95.76	**47.69**	**51.69**	**45.12**	7.43	5.96	6.68	**42.04**
		FAME†	95.69	39.22	40.82	30.63	4.42	3.68	3.03	33.31
		StillMix (Ours)	95.85	43.15	39.29	40.87	**0.07**	**0.01**	**0.00**	41.08
Swin-T	K400	No	**96.21**	37.63	34.37	54.94	3.48	3.02	10.82	36.82
		Mixup	96.17	39.82	40.89	57.79	2.88	3.28	11.62	40.46
		VideoMix	96.00	28.59	37.36	58.26	7.81	11.40	20.60	29.37
		SDN	95.76	34.78	32.56	50.40	2.21	1.42	5.30	36.42
		BE†	96.06	39.76	36.16	56.01	3.55	2.93	10.15	38.62
		ActorCutMix†	95.87	51.02	**55.28**	**69.53**	8.00	8.43	19.32	46.87
		FAME†	95.81	40.62	44.56	37.54	5.74	6.50	6.84	35.14
		StillMix (Ours)	96.02	**55.22**	53.68	65.75	**2.40**	**2.16**	**5.76**	**54.90**
TSM	ImageNet	No	84.84	13.89	8.73	9.58	7.89	4.76	6.21	6.13
		Mixup	86.72	27.52	25.96	24.47	9.83	8.22	10.60	17.88
		VideoMix	83.90	29.33	27.77	23.60	12.12	12.76	13.31	17.12
		SDN	80.41	10.11	6.74	6.82	3.44	2.37	2.26	5.44
		BE†	84.42	14.03	8.29	8.48	8.70	4.67	5.69	5.64
		ActorCutMix†	82.42	**47.60**	**51.00**	**48.84**	20.47	22.33	25.29	**28.48**
		FAME†	83.03	22.95	22.35	13.20	10.38	8.74	5.76	12.70
		StillMix (Ours)	**87.29**	28.24	20.98	25.99	**0.42**	**0.30**	**1.21**	24.60
SlowFast	ImageNet	No	80.82	15.14	11.37	8.91	6.63	3.90	3.45	8.15
		Mixup	83.54	20.95	18.40	16.53	6.75	5.62	5.63	13.56
		VideoMix	81.26	20.09	19.90	19.88	7.93	8.21	9.17	14.01
		SDN	78.07	13.44	8.76	8.37	5.49	2.99	2.79	6.65
		BE†	81.51	16.36	11.99	8.45	6.72	4.11	3.07	8.55
		ActorCutMix†	81.54	**30.71**	**28.50**	21.63	8.38	6.61	6.18	**20.48**
		FAME†	80.82	22.37	23.09	15.83	7.11	6.68	4.24	15.54
		StillMix (Ours)	**84.96**	20.42	17.15	**21.77**	**0.01**	**0.01**	**0.07**	19.76
Swin-T	ImageNet	No	88.20	19.24	16.97	22.01	8.76	7.45	10.53	11.81
		Mixup	88.34	25.73	24.55	33.16	7.37	5.94	10.03	20.85
		VideoMix	88.45	33.28	42.35	44.98	17.31	23.16	24.02	21.63
		SDN	85.75	13.60	11.45	20.03	6.91	5.04	9.91	9.36
		BE†	87.80	19.30	16.31	20.64	9.06	7.09	9.09	11.38
		ActorCutMix†	88.73	**55.59**	**59.83**	**59.88**	23.30	29.87	29.48	**32.77**
		FAME†	86.00	27.41	30.62	21.05	7.13	8.50	4.95	20.11
		StillMix (Ours)	**88.92**	32.16	31.31	36.91	**1.08**	**1.54**	**1.66**	32.14

Table S11: Action recognition accuracy (%) of different methods on the Kinetics-400, Kinetics400-SCUBA and Kinetics400-SCUFO datasets.

Method		Pretraining	Debiasing	Kinetics400	Kinetics400-SCUBA (↑)			Kinetics400-SCUFO (↓)			Contra. Acc. (↑)
					Place365	VQGAN-CLIP	Sinusoid	Place365	VQGAN-CLIP	Sinusoid	
Supervised Action Recognition Models	TSM	ImageNet	-	71.13	40.44	37.39	34.34	18.55	16.61	16.49	22.80
			StillMix	71.28	43.31	40.97	37.15	6.27	5.17	4.24	36.07
	SlowFast	ImageNet	-	65.63	36.96	35.54	34.88	21.74	20.19	20.51	18.98
			StillMix	65.65	37.63	35.71	35.46	14.78	13.35	12.70	25.01
	Swin-T	ImageNet	-	73.95	40.81	39.67	44.75	17.89	15.89	20.73	25.93
			StillMix	73.86	43.44	42.81	46.05	4.76	4.37	7.41	39.41
Debiasing	FAME	K400	-	70.95	37.10	36.34	38.20	18.15	16.10	17.01	23.14
Self-supervised	VideoMAE	K400	-	80.00	50.68	49.41	57.26	23.41	22.65	27.98	29.76
Multi-modal	X-CLIP	Web+K400	-	84.13	53.55	55.53	59.26	32.14	32.73	35.55	25.34

Table S12: Action recognition accuracy (%) of different methods on the HMDB51, HMDB51-SCUBA and HMDB51-SCUFO datasets. K400 denotes Kinetics-400. Mini-K200 denotes Mini-Kinetics-200 [22]. † denotes zero-shot classification.

Method		Pretrain	Debiasing	HMDB51	HMDB51-SCUBA (↑)			HMDB51-SCUFO (↓)			Contra. Acc. (↑)
					Place365	VQGAN-CLIP	Sinusoid	Place365	VQGAN-CLIP	Sinusoid	
Supervised Action Recognition Models	TSM	ImageNet	-	47.56	19.39	16.99	8.49	11.78	11.12	6.41	6.50
			StillMix	54.66	39.52	38.41	33.00	6.98	6.09	3.58	33.14
		K400	-	70.39	45.09	42.16	26.84	23.26	20.03	14.40	22.02
			StillMix	71.52	53.91	52.66	38.13	11.63	8.29	5.38	42.05
	SlowFast	ImageNet	-	47.65	21.24	16.85	10.43	17.84	15.63	11.76	5.26
			StillMix	51.53	33.55	32.14	25.81	14.34	13.68	12.59	21.21
		K400	-	76.25	47.36	48.23	35.00	23.26	23.14	18.78	28.37
			StillMix	76.52	48.31	47.75	40.31	17.76	17.51	15.65	35.20
	Swin-T	ImageNet	-	53.62	23.45	19.39	18.24	18.25	14.80	15.94	6.56
			StillMix	55.36	34.73	30.84	30.83	16.13	11.60	13.20	21.40
		K400	-	73.92	47.61	42.77	41.41	20.68	17.90	22.80	27.84
			StillMix	74.82	53.27	52.43	49.73	13.39	12.66	14.13	40.28
Debiasing Pretraining	SDN	Mini-K200	-	56.60	26.76	23.48	11.13	14.80	14.69	4.96	10.83
	FAME	K400	-	61.10	31.45	28.67	25.12	13.28	13.67	12.93	17.21
Self-supervised	VideoMAE	HMDB51	-	62.60	23.24	27.19	18.55	10.58	10.43	10.94	15.01
Multi-modal	X-CLIP†	Web+K400	-	49.67	22.50	25.47	27.03	18.52	20.31	21.37	9.31

from the video and replace the background for training to mitigate background bias. We directly use the available pretrained checkpoints for evaluation.

Self-supervised Pretraining Method: VideoMAE [20], a strong self-supervised learner with masked autoencoder.

Multi-modal Pretraining Model: X-CLIP [15], an expanded language-image pretrained model with a video-specific prompting scheme.

Table S11, S12 and S13 compare the IID and OOD performance of different pretraining methods on Kinetics-400, HMDB51 and UCF101, respectively. From the results we make the following observations:

Pretraining on large video datasets is by itself an effective method to debias action representations. By comparing the performance of using ImageNet and Kinetics-400 as pretraining datasets in Table S12 and S13, we observe that K400-pretrained models improve the performance on both IID test and SCUBA without too much performance sacrifice on SCUFO. The results demonstrate that pretraining on large video datasets is by itself an effective method to debias action representations. We hypothesize that the size of Kinetics-400 is so large that it contains reasonably

Table S13: Action recognition accuracy (%) of different methods on the UCF101, UCF101-SCUBA and UCF101-SCUFO datasets. K400 denotes Kinetics-400. Mini-K200 denotes Mini-Kinetics-200 [22]. † means zero-shot classification.

Method		Pretraining	Debiasing	UCF101	UCF101-SCUBA (↑)			UCF101-SCUFO (↓)			Contra. Acc. (↑)
					Place365	VQGAN-CLIP	Sinusoid	Place365	VQGAN-CLIP	Sinusoid	
Supervised Action Recognition Models	TSM	ImageNet	-	84.84	13.89	8.73	9.58	7.89	4.76	6.21	6.13
			StillMix	87.29	28.24	20.98	25.99	0.42	0.30	1.21	24.60
		K400	-	94.62	26.79	22.66	27.36	5.63	4.12	2.89	21.83
			StillMix	94.30	37.40	33.85	40.30	0.97	0.81	0.60	36.47
	SlowFast	ImageNet	-	80.82	15.14	11.37	8.91	6.63	3.90	3.45	8.15
			StillMix	84.96	20.42	17.15	21.77	0.02	0.01	0.07	19.76
		K400	-	95.96	34.34	31.00	30.19	2.44	1.51	1.14	30.25
			StillMix	95.85	43.15	39.29	40.87	0.07	0.01	0.00	41.08
	Swin-T	ImageNet	-	88.20	19.24	16.97	22.01	8.76	7.45	10.53	11.81
			StillMix	88.92	32.16	31.31	36.91	1.08	1.54	1.66	32.14
		K400	-	96.21	37.63	34.37	54.94	3.48	3.02	10.82	36.82
			StillMix	96.02	55.22	53.68	65.75	2.40	2.16	5.76	54.90
Debiasing Pretraining	SDN	Mini-K200	-	84.17	10.31	7.82	8.59	1.85	1.47	1.69	7.74
	FAME	K400	-	88.60	18.79	19.06	15.80	1.25	1.21	1.27	17.13
Self-supervised	VideoMAE	UCF101	-	91.30	19.10	18.77	19.38	0.59	0.66	1.03	18.50
Multi-modal	X-CLIP†	Web+K400	-	74.52	24.64	28.44	37.36	16.24	16.88	20.42	15.27

balanced static cues. However, collecting, annotating, and training on large-scale datasets are still costly, while simple augmentations could mitigate static bias even for K400-pretrained models; the minimum improvement of *Contra. Acc.* is 6.83%.

Debiasing pretraining does not mitigate static bias effectively. SDN and FAME adopt debiasing pretraining on large datasets and finetuning on small datasets. In Table S12 and S13, SDN obtains comparable performance on IID and OOD test with ImageNet-pretrained models, though it is pretrained on Mini-Kinetics-200. FAME lags behind K400-pretrained models, though it is also pretrained on Kinetics-400. The results indicate that vanilla supervised pretraining is more effective than debiasing pretraining at mitigating static bias. Effective debiasing pretraining deserves further research attention.

Self-supervised models and multi-modal pretraining models are still vulnerable to static bias. As powerful video representation learners, VideoMAE and X-CLIP obtain good performance on IID tests and SCUBA, but the performance on SCUFO and the *Contra. Acc.* is worse than ImageNet-pretrained Swin-T trained with StillMix. The results indicate that they could not effectively mitigate foreground static bias and learn robust action features. How to improve the robustness of action representations through self-supervised pretraining and prompting large language-vision pretrained models still deserves exploration.

S1.10. Grad-CAM Visualization

In Figure S4, we visualize the Grad-CAM [16] on some videos from UCF101. Trained with StillMix, TSM focuses on the regions with motion. For example, in the second column, the action is "Juggling Balls". TSM trained without StillMix focuses on the background like the grass field and the trees. However, the model trained with StillMix learns to focus on the hand motion. The visualization results validate that StillMix helps to learn motion representations and mitigate reliance of background static cues.

S2. Construction Details of SCUBA and SCUFO

We synthesize SCUBA and SCUFO videos using videos in the test set of the first split of HMDB51 [11] and UCF101 [17], and the validation set of Kinetics-400 [1].

S2.1. Foreground Masks

The details of collecting and producing foreground masks of the three datasets are described as follows.

HMDB51. We use human-annotated segmentation masks of people for 21 action classes from the JHMDB dataset [10]. There are totally 256 videos in the test set of the first split having mask annotations.

UCF101. We use human-annotated bounding boxes of people for 24 action classes provided by the Thumos challenge [9]. There are totally 910 videos in the test set of

Figure S4: Grad-CAM visualization on videos from UCF101 dataset. The first row shows results of TSM trained without StillMix. The second row shows the results of TSM + StillMix. StillMix helps to focus on the motion regions.

the first split having bounding box annotations.

Kinetics-400. We decode the validation videos into frames (we use 15 fps as the frame rate) to extract the foreground mask of each frame. Since there is no available human annotation of Kinetics-400, we use video semantic segmentation model VSS-CFFM [19] and video salient object segmentation model UFO [18] to extract foregrounds. In each frame, we extract the human mask from VSS-CFFM and the salient object mask from UFO, and each of them is smoothed using the masks in three adjacent frames, *i.e.*, the union of the three masks is used as the smoothed mask. The foreground mask is the union of the smoothed human mask and the smoothed salient object mask. The videos in which more than 10% of frames having small foreground masks (*i.e.*, the area of the foreground mask is smaller than 10% of the area of the whole frame) are discarded. Finally, we use the remaining 10,190 videos in the validation set to construct benchmarks.

S2.2. Background Images

The details of generating background images by VQGAN-CLIP [4] and sinusoidal functions are described as follows.

VQGAN-CLIP. 2,000 background images of artistic style are generated by VQGAN-CLIP. Each image is generated from a sentence with the template: "A painting / sketch / illustration / photograph of *scene_name* in the style of *style_name*". In the template, the *scene_name* is the name of a random scene category in Place365 [23]; the *style_name* is a random artistic style sampled from a list: {"Art Nouveau", "Camille Pissarro", "Michelangelo Caravaggio", "Claude Monet", "Edgar Degas", "Edvard Munch", "Fauvism", "Futurism", "Impressionism", "Picasso", "Pop Art", "Modern art", "Surreal Art", "Sandro Botticelli", "Oil Paints", "Water Colours", "Weird Bananas", "Strange Colours"}.

Sinusoid. Each stripe in the stripe images are defined by sinusoidal functions $y = Asin(\omega x + \phi)$. The ranges of each parameter are defined as follows:

- $0 \leq \omega \leq 0.5 \times \pi$
- $-100 \times \omega \leq \phi \leq 100 \times \omega$
- $10 \leq A \leq 110$

The ranges of stripe widths sw and space between two adjacent stripes sp are defined as follows:

- $5 \leq sw \leq 20$
- $3 \times sw \leq sp \leq 5 \times sw$

Each stripe image is generated by uniformly sampling these parameters from the corresponding ranges. The colors of the stripe areas and the background areas are also randomly chosen. After a stripe image is generated, we further rotate it by a random angle and crop the central area of 224×224 as the final stripe image.

S2.3. Synthetic Videos

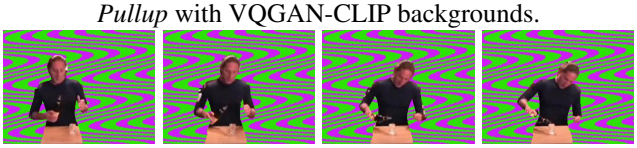
In Figure S5, we show some examples of SCUBA videos. From the shown examples, we observe that the action information is reserved although the backgrounds

Diving with Place365 backgrounds.

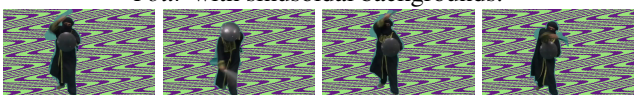
Wrestling with Place365 backgrounds.

GolfSwing with VQGAN-CLIP backgrounds.

Pullup with VQGAN-CLIP backgrounds.



Pour with sinusoidal backgrounds.



Sword Fighting with sinusoidal backgrounds.

Figure S5: Examples of the synthetic videos.

of the videos are replaced, so that the actions in the synthetic videos are recognizable. From each background image source, we provide two SCUBA videos in the data appendix for readers' reference.

S2.4. Human Assessment

We verify that the actions in SCUBA videos can be recognized by human on Amazon Mechanical Turk (AMT). From the same original video, we randomly sample one synthetic video for assessment. Totally, we have 256 synthetic videos in HMDB51-SCUBA, 910 synthetic videos in UCF101-SCUBA and 10,190 synthetic videos in Kinetics400-SCUBA to be assessed.

The AMT workers are asked to determine whether the moving parts in the videos show the labeled action. Figure S6 shows the AMT interface of the assessment task. The interface shows the instruction of the task to workers: "Inspect the full video carefully, and determine whether a specific action is shown in the video. Please determine the actions based on the moving parts instead of the backgrounds, since the backgrounds of some videos are deliberately altered." It also displays a video and a corresponding question: "Do the moving parts of the video show the action *action_name*?" The *action_name* is the name of the provided action label corresponding to the video. The workers are given three options to select: yes, no, and can't tell.

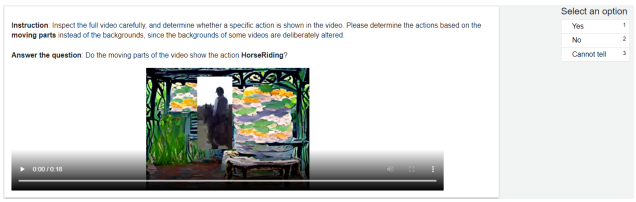


Figure S6: Interface of AMT tasks.

We divide the videos into a number of groups for assessment. In each group, we create the following questions:

- Experimental group (47.5% of the total questions): contains synthetic videos with correct labels.
- Control group (47.5% of the total questions): contains synthetic videos with random incorrect labels. The control group is constructed to prevent the workers from always answering yes to synthetic videos.
- Control questions (5% of the total questions): contain original videos, half of which are assigned correct labels and the other are assigned incorrect labels. The control questions are used to detect random clicking.

Each question is assigned to three different workers to answer. We accept the answers of a worker only if he or she satisfies the following criteria:

- Answered more than one control questions and reached at least 75% of accuracy on the answered control questions.
- Reached at least 90% of accuracy on the answered questions in the control group, in which the synthetic videos are assigned incorrect labels. If a worker does not reach high accuracy on these questions, he or she may tend to answer yes to synthetic videos, which affects the assessment results.

The final answer for each question is obtained by majority voting.

According to the collected answers, the AMT workers were able to recognize the correct action in 876 videos out of 910 UCF101-SCUBA videos (96.15%), 222 videos out of 256 HMDB51-SCUBA videos (86.33%) and 8681 videos out of 10190 Kinetics400-SCUBA videos (85.19%).

S3. Implementation Details

S3.1. Datasets

UCF101 [17] has 13,320 web videos recorded in unconstrained environments, belonging to 101 classes. We use

Table S14: The hyper-parameters for training TSM, SlowFast, Swin-T without data augmentation or debiasing techniques. UCF, HMDB and K400 denote UCF101, HMDB51 and Kinetics-400, respectively.

Hyper-parameter	ImangeNet-pretrained									Kinetics-pretrained					
	TSM			SlowFast			Swin-T			TSM		SlowFast		Swin-T	
	UCF	HMDB	K400	UCF	HMDB	K400	UCF	HMDB	K400	UCF	HMDB	UCF	HMDB	UCF	HMDB
frames per video	8	8	8	64	64	64	32	32	32	8	8	64	64	32	32
epoch	100	100	50	100	100	50	30	30	30	25	25	25	25	30	30
optimizer	SGD			SGD			AdamW			SGD		SGD		AdamW	
linear warmup epochs	-	-	-	-	-	-	2.5	2.5	2.5	-	-	-	-	2.5	2.5
base learning rate	0.0025	0.01	0.0025	0.0025	0.005	0.01	0.002	0.005	0.001	0.0025	0.005	0.005	0.005	0.0005	0.0005
learning rate schedule	$\times 0.1$ at 40% and 80% of the total epochs						cosine			$\times 0.1$ at 10^{th} and 20^{th} epoch				cosine	
weight decay	0.001	0.01	0.0001	0.001	0.01	0.0001	0.01	0.02	0.02	0.001	0.01	0.001	0.01	0.01	0.05

Table S15: The hyper-parameters for training action recognition models with different video data augmentation and debiasing methods. UCF, HMDB and K400 denote UCF101, HMDB51 and Kinetics-400, respectively.

Augmentation or Debiasing	Hyper-parameter	ImangeNet-pretrained									Kinetics-pretrained					
		TSM			SlowFast			Swin-T			TSM		SlowFast		Swin-T	
		UCF	HMDB	K400	UCF	HMDB	K400	UCF	HMDB	K400	UCF	HMDB	UCF	HMDB	UCF	HMDB
Mixup	base learning rate	0.01	0.005	0.005	0.005	0.005	0.01	0.005	0.005	0.001	0.0025	0.005	0.005	0.005	0.0005	0.0005
	weight decay	0.001	0.01	0.0001	0.001	0.01	0.0001	0.01	0.02	0.02	0.001	0.01	0.001	0.01	0.01	0.05
	P_{aug}	1.0	0.75	1.0	1.0	0.25	1.0	1.0	0.75	1.0	0.25	0.25	0.5	0.25	0.5	0.75
	$Beta(\alpha, \beta)$	$(0.2, 0.2)$														
VideoMix	base learning rate	0.01	0.01	0.005	0.005	0.005	0.01	0.002	0.005	0.001	0.005	0.005	0.005	0.005	0.0002	0.0005
	weight decay	0.001	0.01	0.0001	0.001	0.01	0.0001	0.01	0.02	0.02	0.001	0.01	0.001	0.01	0.01	0.05
	P_{aug}	1.0	0.25	0.75	0.5	0.5	1.0	0.75	0.75	0.75	0.25	0.5	0.25	0.25	0.5	0.25
	$Beta(\alpha, \beta)$	$(1.0, 1.0)$														
SDN	base learning rate	0.02	0.02	0.01	0.02	0.02	0.01	0.002	0.002	0.001	0.002	0.01	0.02	0.01	0.0001	0.0002
	weight decay	0.001	0.01	0.0001	0.001	0.01	0.0001	0.01	0.02	0.02	0.001	0.01	0.001	0.01	0.01	0.05
BE	base learning rate	0.0025	0.005	0.005	0.005	0.01	0.01	0.002	0.005	0.001	0.0025	0.005	0.005	0.005	0.0005	0.0005
	weight decay	0.001	0.01	0.0001	0.001	0.01	0.0001	0.01	0.02	0.02	0.001	0.01	0.001	0.01	0.01	0.05
	P_{aug}	0.75	0.75	0.25	1.0	0.5	0.75	0.75	0.75	0.5	0.25	0.5	0.25	0.5	0.5	0.25
ActorCutMix	base learning rate	0.005	0.01	0.01	0.01	0.005	0.01	0.002	0.005	0.001	0.0025	0.005	0.005	0.005	0.0001	0.0005
	weight decay	0.001	0.01	0.0001	0.001	0.01	0.0001	0.01	0.02	0.02	0.001	0.01	0.001	0.01	0.01	0.05
	P_{aug}	0.75	0.5	0.25	0.25	0.5	0.25	0.5	0.75	0.25	0.25	0.25	0.5	0.5	0.5	0.25
FAME	base learning rate	0.0025	0.01	0.005	0.0025	0.005	0.005	0.005	0.005	0.001	0.0025	0.005	0.005	0.005	0.0005	0.0005
	weight decay	0.001	0.01	0.0001	0.001	0.01	0.0001	0.01	0.02	0.02	0.001	0.01	0.001	0.01	0.01	0.05
	P_{aug}	0.25	0.75	0.25	0.25	0.5	0.25	0.25	0.5	0.25	0.25	0.25	0.25	0.25	0.25	0.5
StillMix	base learning rate	0.005	0.005	0.005	0.0025	0.005	0.005	0.002	0.005	0.001	0.0025	0.005	0.005	0.005	0.0005	0.0005
	weight decay	0.001	0.01	0.0001	0.001	0.01	0.0001	0.01	0.02	0.02	0.001	0.01	0.001	0.01	0.01	0.05
	P_{aug}	0.25	0.5	0.25	0.75	0.25	0.125	0.75	0.5	0.25	0.25	0.25	0.5	0.25	0.25	0.75
	τ	25	15	25	15	25	25	50	50	50	10	50	10	75	50	10
	frame bank size	4096														
	$Beta(\alpha, \beta)$	$(200, 200)$	$(200, 200)$	$(20, 20)$	$(200, 200)$	$(200, 200)$	$(2, 2)$	$(200, 200)$	$(200, 200)$	$(20, 60)$	$(200, 200)$		$(200, 200)$		$(200, 200)$	

the first official train-test split in our experiments and report the performance on the test set.

HMDB51 [11] consists of 51 classes and 6,766 videos extracted from a variety of sources ranging from digitized movies to YouTube videos. We use the first official train-test split and report the performance on the test set.

Kinetics-400 [1] contains more than 250k videos in 400 classes. We train the models on the training set (around 240k videos) and reported performance on the validation set (around 20k videos) as in prior works [13, 7, 14, 21].

S3.2. Action Recognition Models

For TSM [13], we use ResNet-50 as the backbone. For SlowFast [7], we use 3D ResNet-50 with filters inflated from 2D to 3D [1] as the backbone. And we use the version of 4×16 ($T \times \tau$) in our experiments. For Video Swin Transformer [14], we use the tiny version (denoted as Swin-T) in our experiments.

S3.3. Computational Resources

Our experiments are conducted on GPU clusters (containing Tesla V100, Tesla P100, GeForce RTX 3090, RTX A6000) with the PyTorch codebase MMAction2 [3].

S3.4. Training the Reference Network of StillMix

We train the Reference Network $\mathcal{R}$ of StillMix with the following settings:

- Network: ResNet-50, SlowFast-2D (ResNet-50 as backbone), tiny Swin Transformer
- Pretrained: ImageNet
- Optimizer: SGD

- Base learning rate: 0.01. The base learning rate corresponds to the batch size of 64. We apply the Linear Scaling Rule [8] to set the learning rate according to the real batch sizes.
- Epochs: 50
- Learning rate schedule: learning rate is divided by 10 at the 20^{th} and 40^{th} epoch
- Weight decay: 0.00001

S3.5. Training the Main Network

Random Seeds. On UCF101 and HMDB51, for each model and each data augmentation or debiasing method, we fix the random seeds as 1, 2, and 3 to conduct three times of training. The reported accuracies are the mean accuracies of the three runs. On Kinetics-400, we fix the random seeds as 1 and conduct only one time of training.

Other Data Augmentations. Except the data augmentation methods discussed in the main paper, we also use two commonly used data augmentations for each model during training: *(1)* The shorter ends of video frames are resized to 256 and an area of 224×224 is randomly cropped. *(2)* Each video is flipped horizontally with a probability of 0.5.

Hyper-parameters. On UCF101 and HMDB51, we randomly sample 20% of the training samples to form a validation set for hyper-parameter tuning. On Kinetics-400, we randomly sample 50% of the training samples to form a training-validation split to tune hyper-parameters. In the training-validation split, the proportion of training and validation samples is 19:1. After the best hyper-parameters are selected, we train the models on the full training set with the best hyper-parameters.

For action recognition models without data augmentation or debiasing methods applied, we tune the base learning rate and the weight decay on the validation set and fix other hyper-parameters as pre-defined values. The base learning rate corresponds to the batch size of 64, and we apply the Linear Scaling Rule [8] to set the learning rate according to the real batch sizes. For data augmentation methods, we additionally tune the augmentation probability P_{aug} on the validation set. Table S14 and S15 show the hyper-parameters we used in different action recognition models as well as the data augmentation and debiasing methods.

S3.6. Evaluation

The checkpoint at the last epoch is used for evaluation. Given a video, we resize the shorter ends of frames to 256 and use a center crop of 224×224 from a single clip for evaluation.

References

[1] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4733. IEEE, 2017. 9, 12

[2] Jinwoo Choi, Chen Gao, Joseph CE Messou, and Jia-Bin Huang. Why can't i dance in the mall? learning to mitigate scene bias in action recognition. In *Advances in Neural Information Processing Systems*, 2019. 5

[3] MMAction2 Contributors. Openmmlab's next generation video understanding toolbox and benchmark. https://github.com/open-mmlab/mmaction2, 2020. 12

[4] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance. *arXiv preprint arXiv:2204.08583*, 2022. 10

[5] Shuangrui Ding, Maomao Li, Tianyu Yang, Rui Qian, Haohang Xu, Qingyi Chen, Jue Wang, and Hongkai Xiong. Motion-aware contrastive video representation learning via foreground-background merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9716–9726, June 2022. 1, 5

[6] Haodong Duan, Yue Zhao, Kai Chen, Yuanjun Xiong, and Dahua Lin. Mitigating representation bias in action recognition: Algorithms and benchmarks. *arXiv preprint arXiv:2209.09393*, 2022. 2

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *IEEE international conference on computer vision*, pages 6202–6211, 2019. 12

[8] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 13

[9] H. Idrees, A. R. Zamir, Y. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017. 9

[10] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *IEEE international conference on computer vision*, pages 3192–3199, 2013. 9

[11] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *International conference on computer vision*, pages 2556–2563. IEEE, 2011. 9, 12

[12] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Limin Wang, and Yu Qiao. Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer. *arXiv preprint arXiv:2211.09552*, 2022. 3

[13] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *IEEE International Conference on Computer Vision*, pages 7082–7092. IEEE, 2021. 12

[14] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021. 12

[15] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, pages 1–18. Springer, 2022. 8

[16] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE international conference on computer vision*, pages 618–626, 2017. 9

[17] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 9, 11

[18] Yukun Su, Jingliang Deng, Ruizhou Sun, Guosheng Lin, and Qingyao Wu. A unified transformer framework for group-based segmentation: Co-segmentation, co-saliency detection and video salient object detection. *arXiv preprint arXiv:2203.04708*, 2022. 10

[19] Guolei Sun, Yun Liu, Henghui Ding, Thomas Probst, and Luc Van Gool. Coarse-to-fine feature mining for video semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3126–3137, 2022. 10

[20] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv preprint arXiv:2203.12602*, 2022. 8

[21] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1895–1904, 2021. 12

[22] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. *arXiv preprint arXiv:1712.04851*, 1(2):5, 2017. 8, 9

[23] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. 10

[24] Yuliang Zou, Jinwoo Choi, Qitong Wang, and Jia-Bin Huang. Learning representational invariances for data-efficient action recognition. *arXiv preprint arXiv:2103.16565*, 2021. 1