Reinforcement Learning meets Masked Video Modeling : Trajectory-Guided Adaptive Token Selection

Supplementary Material

1. Datasets

Something-Something V2 (SSv2) [3] is a curated video dataset for human action classification, comprising 174 classes and a total of 220,847 videos. Each video depicts a single action with a duration ranging from 2 to 6 seconds. SSv2 is a motion-focused dataset, where temporal relationships are more pronounced compared to other datasets.
Kinetics-400 (K400) [4] is a widely used large-scale video dataset, comprising 400 classes and over 250,000 videos. Each video, approximately 10 seconds in duration, captures a single action.
HMDB51 [5] comprises 51 classes and a total of 6,766 videos. HMDB51 emphasizes appearance information over motion dynamics.
UCF101 [6] comprises 13,320 video clips categorized into 101 classes. These classes span five activity types: body motion, human-to-human interaction, human-to-object interaction, musical instrument performance, and sports.

2. Additional Implementation Details

2.1. Data Preprocessing

Our data processing pipeline closely follows AdaMAE [1] for pre-training. We extract 16 frames of dimension 224×224 from the videos, using a temporal stride of 4 (K400) and 2 (HMDB51/UCF101/SSv2), with the starting frame randomly selected [2]. During pre-training, we apply data augmentation techniques, including random resized cropping in the spatial domain, random scaling within the range $\in [0.5, 1]$, and random horizontal flipping [2].

Table 1. Hyperparameter setting for pre-training across all benchmark datasets.

Configuration	Value
Learning rate for g_θ - lp	1.5e-6
Epochs to train f_ϕ only - m_o	10
Steps to train f_ϕ and record g_θ episodes - k	1
Softmax Temperature	1
Policy loss coefficient - c_1	1e-4
Value loss coefficient - c_2	1e-4
Entropy coefficient - c_3	1e-4
Optimizer	AdamW
Optimizer betas	0.9, 0.95
Batch size	32
Base learning rate	1.5e-4
Learning rate schedule	cosine decay
Warmup epochs	40
Augmentation	MultiScaleCrop

Table 2. Hyperparameter (m_o, k) tuning for pre-training, evaluated based on reconstruction error on UCF101 and HMDB51. Same configuration is adopted for SSv2 and K400 as in UCF101. Best configuration is shown in gray.

(m_o, k)	UCF101	HMDB51
(0, 1)	0.5211	0.8051
(1, 1)	0.5205	0.8195
(5, 1)	0.5304	0.8535
(10, 1)	0.5135	0.8278
(25, 1)	0.5269	0.8987
(100, 1)	0.6662	0.9291
(50, 5)	0.7735	0.9772
(50, 10)	0.8149	0.9776
(50, 25)	0.9201	-

2.2. Hyper-parameter Setting

Pre-training. The hyperparameter configurations used during the pre-training phase across all benchmark datasets are presented in Table 1. For (m_o, k), hyperparameter tuning is conducted on the UCF101 and HMDB51 datasets (Table 2), and the configuration that minimizes the reconstruction error is selected. Similarly we also perform hyperparameter tuning for coefficients (c_1,c_2,c_3) in Table 3 during pretraining on UCF101 and observe that (1e-4, 1e-4, 1e-4) minimizes the reconstruction error. Empirical observations indicate that the optimal configuration for UCF101 also performs effectively on subset of K400 and SSv2 (small scale pre-training setup). It is to be noted that we use reconstruction loss for tuning these hyper-parameters because behaviour of reconstruction loss during pretraining is more interpretable in terms of convergence than the sampling loss.
Fine-tuning. The hyperparameter setting for end-to-end fine-tuning on the downstream task of action recognition across all benchmarks is summarized in Table 4.

2.3. Encoder-Decoder Architecture

We adopt an asymmetric encoder-decoder architecture [1] for self-supervised pre-training and augment it with *TATS* module and only keep the encoder during the fine-tuning. In particular, the design of the encoder-decoder is based on 16-frame vanilla ViT-Base architecure. Table 5 provides an overview of the encoder-decoder architecture utilized in our framework.

Table 3. Hyperparameter (c_1, c_2, c_3) tuning for pre-training, evaluated based on reconstruction error on UCF101. Same configuration is adopted for SSv2, K400 and HMDB51. (m_o, k) are fixed as $(10, 1)$. Best configuration is shown in gray.

(c_1, c_2, c_3)	UCF101
(1e-4, 1e-3, 1e-3)	0.5188
(1e-4, 1e-3, 1e-4)	0.5167
(1e-4, 1e-4, 1e-3)	0.5246
(1e-3, 1e-4, 1e-4)	0.8482
(1e-4, 1e-4, 1e-4)	0.5135
(1e-5, 1e-4, 1e-4)	0.5239
(1e-3,1e-3,1e-4)	0.5215
(1e-3, 1e-3, 1e-4)	0.7869
(1e-5, 1e-5, 1e-5)	0.5173

Table 4. Hyperparameter setting for end-to-end fine-tuning for all benchmark datasets.

Configuration	Value
Optimizer	AdamW
Optimizer Betas	{0.9, 0.999}
Batch size	8
Weight Decay	5e-2
Base Learning Rate	1e-3
Learning Rate Schedule	cosine decay
Layer-wise learning rate decay	0.75
Warmup epochs	5
RandAug	9, 0.5
Label Smoothing	0.1
Mixup	0.8
CutMix	1.0
DropPath	0.1
# Temporal Clips	5 (k400), 2 (ssv2/hmdb/ucf)
# Spatial Crops	3

3. Linear Probing Evaluation.

Table 6 presents the top-1 and top-5 accuracy obtained after linear probing evaluation of our method across different mask ratios, $\rho = \{0.85, 0.90, 0.95\}$ under the small scale setting. Our method outperforms both AdaMAE [1] and VideoMAE [7] on UCF101, HMDB51, and SSv2 datasets. For Kinetics-400, the performance of our model exceeds that of AdaMAE [1], while being marginally less than VideoMAE [7]. The potential cause for this observation can be associated to reduced number of pretraining epochs under the small scale setting.

4. Large Scale Pre-training Results

We conduct pre-training (800 epochs) and finetuning (100 epochs) of our model on full SSv2 [3] dataset for $\rho = 0.95$ on 8 Nvidia A100 GPUs. In order to ensure fairness in comparison, we also pre-train (800 epochs) and finetune (100 epochs) both baselines VideoMAE [7] and AdaMAE [1] on full SSv2 for $\rho = 0.95$ with the same GPU setup using their public source code and default configuration[1] .

Table 7 presents the top-1 and top-5 accuracy obtained in this experiment. We observe that our approach outperforms both the baselines under aggressive masking setting even for large scale experiments for both 400 and 800 pretraining epochs. This highlights the effectiveness and generalization capability of the proposed *TATS* module and the training strategy in terms of learning a better feature quality than learnt by [1, 7].

Due to the availability of limited computational resources, our experiments in this setup is limited.

Note : The baseline results for large-scale experiments (VideoMAE [7], AdaMAE [1]) are slightly lower than those

[1] GPU setup for reproducing baselines is same as ours. Denoted by †

Table 5. Encoder-Decoder architecture based on AdaMAE [1]. TATS : Trajectory Aware Adaptive Token Sampler. MHA : Multi-Head Self-Attention

Stage	ViT-Base	Output shape
Input Video	stride $4 \times 1 \times 1$ for K400 stride $2 \times 1 \times 1$ for ssv2/ucf/hmdb	$3 \times 16 \times 224 \times 224$
Tokenization	stride $2 \times 16 \times 16$ emb. dim 768 kernel size $2 \times 16 \times 16$	1568×768
Masking	TATS Masking mask ratio ρ	$[(1-\rho) \times 1568] \times 768$
Encoder	$[MHA(768)] \times 12$	$[(1-\rho) \times 1568] \times 768$
Projection	$MHA(384)$ concat masked tokens	1568×384
Decoder	$[MHA(384)] \times 4$	$[(1-\rho) \times 1568] \times 384$
Projector	$MLP(1536)$	1568×1536
Reshaping	from 1536 to $3 \times 2 \times 16 \times 16$	$3 \times 16 \times 224 \times 224$

Table 6. Comparison of **Linear Probing** result of Our model against baselines ([1, 7]) (†[1]) on action recognition task across benchmark datasets and different ρ with top-1/top-5 accuracy as evaluation metric. (↑ / ↓ : denotes increase/decrease in performance)

Dataset	Mask Ratio ρ	VideoMAE† [7] top-1	VideoMAE† [7] top-5	AdaMAE† [1] top-1	AdaMAE† [1] top-5	**Ours** top-1	**Ours** top-5
UCF101	0.85	46.91	75.71	43.72	71.42	48.54 ↑	76.80 ↑
	0.90	47.38	76.35	45.92	73.15	49.28 ↑	77.73 ↑
	0.95	41.81	70.85	46.19	74.31	49.05 ↑	77.22 ↑
HMDB51	0.85	20.57	50.52	21.42	52.28	22.53 ↑	53.91 ↑
	0.90	19.99	50.72	22.66	54.17	22.79 ↑	54.10 ↓
	0.95	17.64	48.37	21.81	51.56	23.50 ↑	52.28 ↑
Kinetics-400	0.85	12.86	31.33	10.51	26.49	11.61 ↓	28.64 ↓
	0.90	14.27	33.64	11.35	27.46	12.68 ↓	30.28 ↓
	0.95	14.75	34.50	13.36	30.90	13.66 ↓	31.72 ↓
SSv2	0.85	08.96	23.43	09.91	25.47	10.29 ↑	26.04 ↑
	0.90	10.27	25.85	11.06	27.19	11.86 ↑	28.70 ↑
	0.95	11.18	27.68	12.75	30.66	13.39 ↑	31.88 ↑

Table 7. **Large Scale Pre-training and Finetuning Results.** Comparison of fine-tuning result of Our model against baselines ([1, 7]) (†[1]) on action recognition task for full SSv2 and $\rho = 0.95$ with top-1/top-5 accuracy as evaluation metric. (↑/↓: denotes increase/decrease in performance)

Method	Pretrain Epochs	top-1	top-5
VideoMAE† [7]$_{\rho=95\%}$	400	65.48	89.33
AdaMAE† [1]$_{\rho=95\%}$	400	65.52	88.67
Ours$_{\rho=95\%}$	400	65.98 (↑)	89.16(↑)
VideoMAE† [7]$_{\rho=95\%}$	800	65.92	89.07
AdaMAE† [1]$_{\rho=95\%}$	800	66.26	88.62
Ours$_{\rho=95\%}$	800	66.55 (↑)	88.84(↓)

reported in their original publications. We attribute this discrepancy to differences in the GPU pretraining setup. Our experiments use a single node with 8 A100 GPUs, whereas the original works used 8 nodes with 8 A100 GPUs each. All results were reproduced using the publicly available code from the original publication.

5. Limitations and Future Work.

Our proposed *TATS* and training recipe further need to be empirically validated on large scale experimental settings, other downstream tasks and extended to other modalities. Furthermore, with the recent resurgence in RL research due to its applications in LLMs, it is important to reconsider strategies that integrate dynamic computation into masked modeling approaches, optimizing them through RL algorithms. We plan to conduct future studies around these topics. We hope this work can motivate further research in this direction.

6. Mask Visualization

Here we show visualizations of adaptive sampling learned by our *TATS* module across benchmark dataset for different mask ratios $\rho = \{0.95, 0.9, 0.85\}$ in Figure 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12.

In all of these Figures, first row represents input video frames, the second row depicts the prediction/reconstruction, the third row shows the reconstruction error, the fourth row represents the probability of sampling the space-time patch, fifth row shows the adaptive masks learned by *TATS*. The last row depicts the binary masks learned by AdaMAE [1] for comparison.

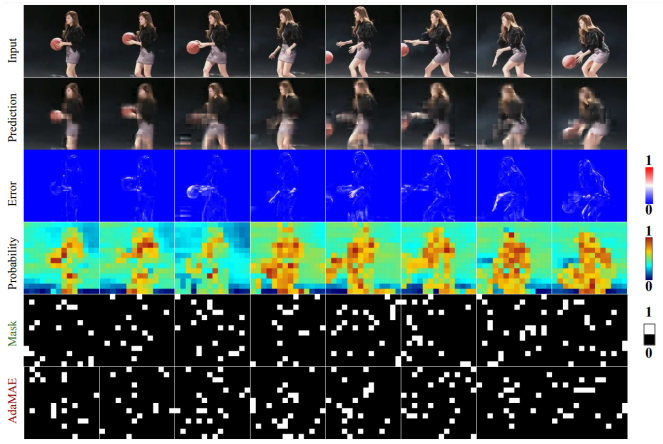

Figure 1. Sample Visualization of a Kinetics 400 video with adaptive sampling using *TATS* with mask ratio $\rho = 0.95$. Compared with AdaMAE [1] masks.

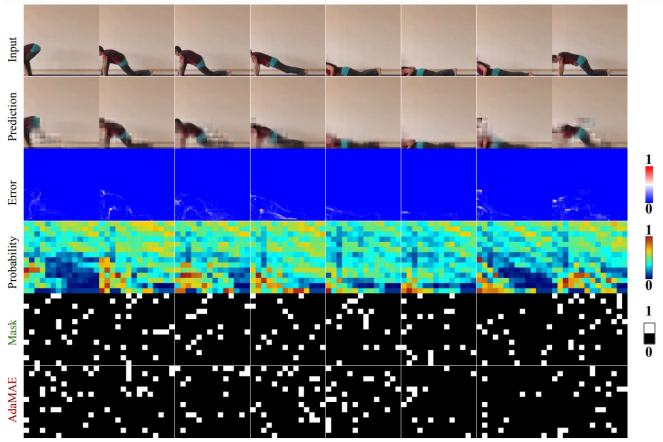

Figure 2. Sample Visualization of a Kinetics 400 video with adaptive sampling using *TATS* with mask ratio $\rho = 0.9$. Compared with AdaMAE [1] masks.

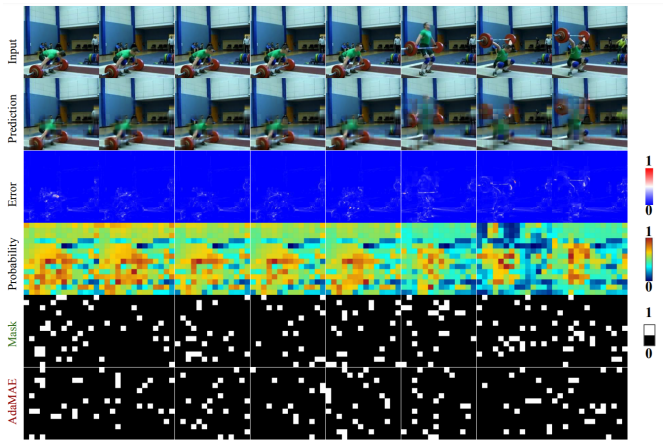

Figure 3. Sample Visualization of a Kinetics 400 video with adaptive sampling using *TATS* with mask ratio $\rho = 0.85$. Compared with AdaMAE [1] masks.

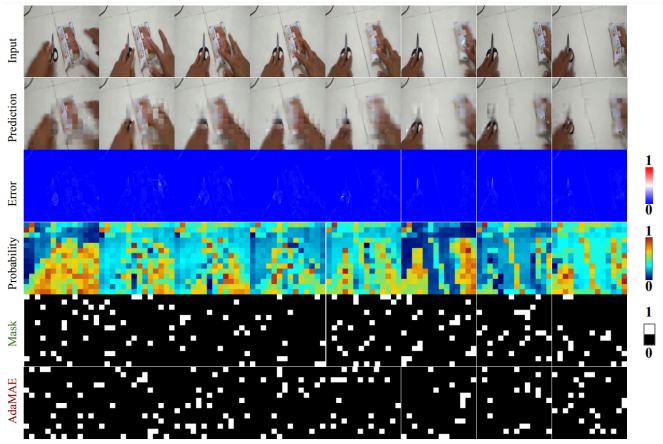

Figure 4. Sample Visualization of a SSv2 video with adaptive sampling using *TATS* with mask ratio $\rho = 0.95$. Compared with AdaMAE [1] masks.

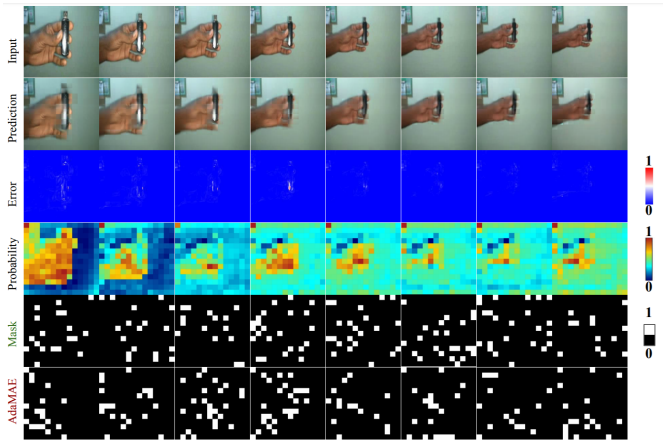

Figure 5. Sample Visualization of a SSv2 video with adaptive sampling using *TATS* with mask ratio $\rho = 0.9$. Compared with AdaMAE [1] masks.

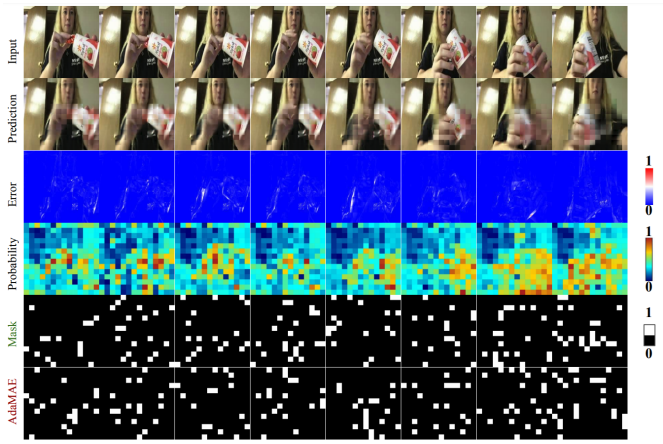

Figure 6. Sample Visualization of a SSv2 video with adaptive sampling using *TATS* with mask ratio $\rho = 0.85$. Compared with AdaMAE [1] masks.

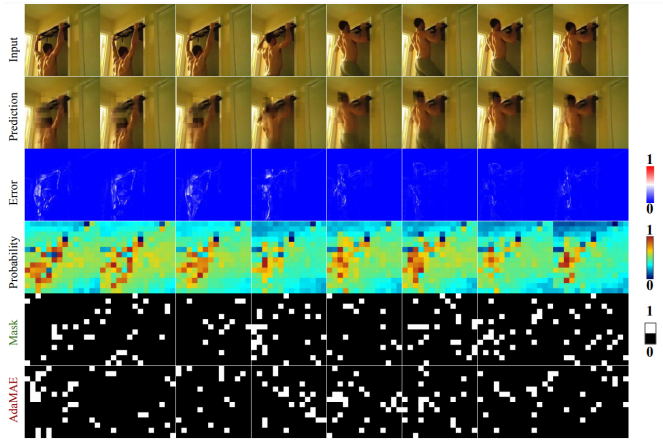

Figure 7. Sample Visualization of a UCF101 video with adaptive sampling using *TATS* with mask ratio $\rho = 0.95$. Compared with AdaMAE [1] masks.

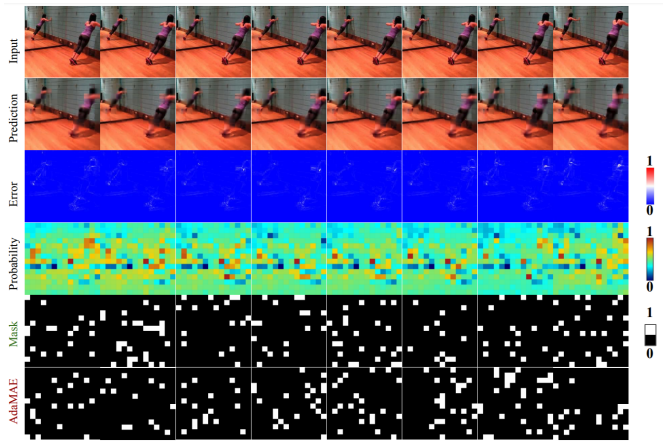

Figure 8. Sample Visualization of a UCF101 video with adaptive sampling using *TATS* with mask ratio $\rho = 0.9$. Compared with AdaMAE [1] masks.

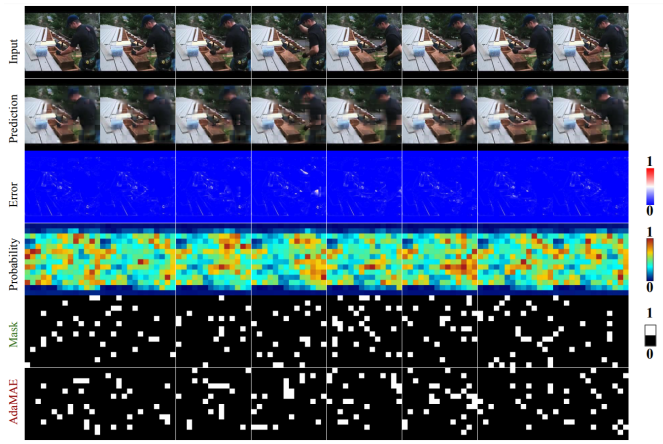

Figure 9. Sample Visualization of a UCF101 video with adaptive sampling using *TATS* with mask ratio $\rho = 0.85$. Compared with AdaMAE [1] masks.

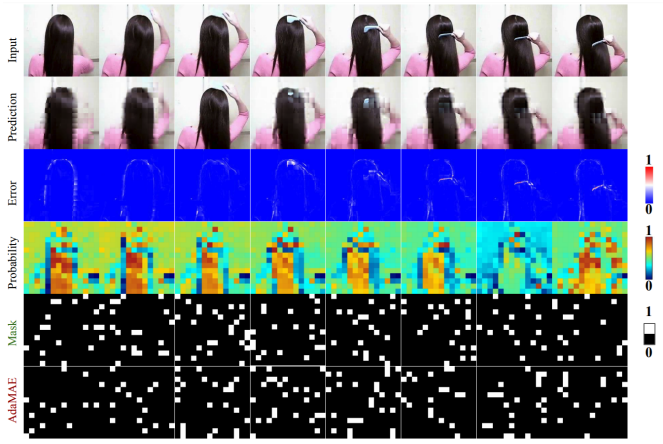

Figure 10. Sample Visualization of a HMDB51 video with adaptive sampling using *TATS* with mask ratio $\rho = 0.95$. Compared with AdaMAE [1] masks.

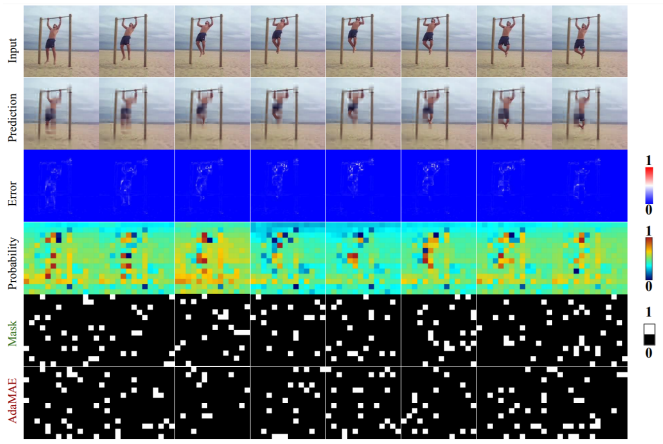

Figure 11. Sample Visualization of a HMDB51 video with adaptive sampling using *TATS* with mask ratio $\rho = 0.9$. Compared with AdaMAE [1] masks.

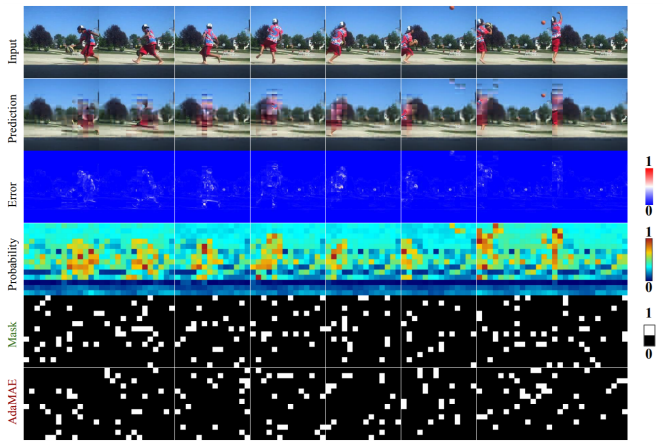

Figure 12. Sample Visualization of a HMDB51 video with adaptive sampling using *TATS* with mask ratio $\rho = 0.85$. Compared with AdaMAE [1] masks.

References

[1] Wele Gedara Chaminda Bandara, Naman Patel, Ali Gholami, Mehdi Nikkhah, Motilal Agrawal, and Vishal M Patel. Adamae: Adaptive masking for efficient spatiotemporal learning with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14507–14517, 2023. 1, 2, 3, 4, 5

[2] Christoph Feichtenhofer, Yanghao Li, Kaiming He, et al. Masked autoencoders as spatiotemporal learners. *Advances in neural information processing systems*, 35:35946–35958, 2022. 1

[3] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017. 1, 2

[4] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1

[5] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE, 2011. 1

[6] 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. 1

[7] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, 2022. 2, 3