Dancing with Still Images:
Video Distillation via Static-Dynamic Disentanglement

Supplementary Material

7. Selection of MiniUCF

We train and test on split 1 of UCF101 on MiniC3D and select the top 50 classes based on accuracy. MiniUCF can reach an accuracy of **57.2%** on MiniC3D. We provide the 50 categories we have selected in Suppl. Tab. 6.

ApplyEyeMakeup	BalanceBeam	BandMarching
BaseballPitch	Basketball	BasketballDunk
Biking	Billiards	BlowingCandles
Bowling	BreastStroke	CleanAndJerk
CliffDiving	CricketShot	Diving
FloorGymnastics	FrisbeeCatch	GolfSwing
HammerThrow	HighJump	HorseRace
HorseRiding	HulaHoop	IceDancing
JumpingJack	Knitting	MilitaryParade
Mixing	ParallelBars	PlayingPiano
PlayingViolin	PoleVault	PommelHorse
Punch	Rafting	Rowing
SkateBoarding	Skiing	Skijet
SkyDiving	SoccerPenalty	StillRings
SumoWrestling	Surfing	Swing
TennisSwing	TrampolineJumping	UnevenBars
VolleyballSpiking	WritingOnBoard	

Table 6. Action classes in the adopted. We highlight the static group with blue and the dynamic group with orange.

8. Ablation Study of Uniformity

We conduct experiments to justify the necessity of uniformity in Supp. Tab. 7. Eight real frames are split evenly or non-evenly into 4 segments and aligned to 4 synthetic frames, among which uniform segmentation achieves the best performance.

Segment Sizes	Acc.	Segment Sizes	Acc.
1,1,5,1	15.8±0.2	1,1,1,5	15.7±0.5
1,1,2,4	14.1±0.2	1,4,1,2	15.1±0.2
3,1,2,2	17.6±0.7	1,2,2,3	16.8±0.1
2,2,2,2 (uniform)	**17.6±0.2**		

Table 7. Results on UCF101 with different uniormity: $N_{real} = 8$ are split unevenly to $N_{syn} = 4$ segments.

SPC	DPC	Acc	S_Acc	D_Acc
1	0	13.7	14.9	13.0
	1	17.5	18.0	16.9
	2	19.6	21.1	17.1
	3	20.6	23.1	19.5
2	1	22.3	23.0	21.4
	2	23.3	24.1	23.1

Table 8. Test accuracies of static and dynamic group on network trained with distilled data by different SPC and DPC for MiniUCF IPC=1. Acc: test accuracies of all classes. S_Acc: test accuracies of the static group. D_Acc: test accuracies of the dynamic group.

9. Impact of Video Dynamics on Distillation

9.1. Selection of Static and Dynamic Group

We utilize a pre-trained four-layer 2D convolutional network to extract features from individual frames. Subsequently, we compute the Hamming distance between the features of consecutive frames. Then, we derive the average inter-frame Hamming distance for each video segment, allowing us to ascertain the average inter-frame Hamming distance for each class. We classify 50% of the classes with smaller average inter-frame Hamming distances into the static group, while the remaining 50% with larger distances are designated as the dynamic group. In Suppl. Tab. 6, we highlight the static and dynamic groups using distinct colors.

9.2. More Detailed Results

We show more detailed results in Suppl. Tab. 8. We can observe that both increasing static memory and dynamic memory concurrently enhance the accuracy of static and dynamic classes. Additionally, we note that when the disparity in quantity between static memory and dynamic memory

	MiniC3D	CNN+GRU	CNN+LSTM	C3D*[3, 21]
UCF101 [19]	33.7	30.4	27.92	51.6
HMDB51 [11]	28.6	24.0	23.2	24.3

Table 9. Top-1 Action Recognition Accuracies on Different Models. For UCF101 and HMDB51, we train and test on split 1. $\star$: The results of C3D are referenced from [3].

becomes larger, there is an imbalance in the accuracy of the static group and dynamic group. The reason for this lies in the random pairing of dynamic memory and static memory during sampling. If there is an excess of dynamic memory, multiple instances of dynamic memory may end up learning the same content during training, essentially augmenting the static information. Therefore, we recommend maintaining a 1:1 ratio between static memory and dynamic memory when utilizing our paradigm.

10. Visualization

10.1. Optical Flows

Our distilled video data could also generalize beyond the classification task. We show some visualized optical flow extraction following [1] in Fig. 10.

10.2. Inter-frame Differences

We show more visualized inter-frame differences of MTT [4] and MTT+Ours for MiniUCF IPC=1 in Fig. 12 and Fig. 13 (last pages).

11. Implementation Details

11.1. Network Structure

In this section, we provide a detailed introduction to models used in the experiments, including MiniC3D, CNN+GRU, and CNN+LSTM. Additionally, we compare their performance with C3D[21] on the UCF101[19] and HMDB51[19] classification tasks.

MiniC3D. Suppl. Fig. 11(a) shows the structure of MiniC3D. For Conv3D blocks except for Conv3D 4, we use 3×7× 7 kernels with 1× 2× 2 strides and 1× 3× 3 paddings. Conv3D 4 is used for classification, which has 1×1× 1 kernel and 1× 1× 1 stride. Channels are denoted below block names in Suppl. Fig. 11. For pooling operations, we employ a 1×2× 2 kernel for max-pooling in pool3D 1, and 2× 2× 2 kernels for max-pooling in both pool3D 2 and pool3D 3. In contrast, we utilize average pooling in pool3D 4.

CNN+GRU. Suppl. Fig. 11(b) shows the structure of CNN+GRU. For Conv blocks, we use 3×3 kernels with 1× 1 strides and 1×1 paddings. Channels are denoted below block names in Suppl. Fig. 11. For pooling operations, we employ 2× 2 kernels for average pooling in both pool 1, pool 2, and pool 3. The GRU block is a single-layer GRU. A linear layer is used for classification.

Dataset	IPC	lr_img	batch_syn	syn_steps
MiniUCF	1	1e5	50	10
	5	1e5	128	5
HMDB51	1	1e4	51	10
	5	1e6	128	5
Kinetics400	1	5e5	256	10
	5	1e7	256	5
SSv2	1	1e5	256	10
	5	1e6	256	5

(a) MTT [4]

Dataset	IPC	lr_dynamic	lr_hal	batch_syn	syn_steps
MiniUCF	1	1e4	1e-3	50	10
	5	1e4	1e-3	128	5
HMDB51	1	1e5	1e-2	51	10
	5	1e6	1e-2	128	5
Kinetics400	1	1e3	1e-2	256	10
	5	1e7	1e-2	256	5
SSv2	1	1e4	1e-2	256	10
	5	1e5	1e-2	256	5

(b) MTT+Ours

Table 10. Hyper-parameters for MTT and MTT+Ours.

CNN+LSTM. Suppl. Fig. 11(c) shows the structure of CNN+LSTM. For Conv blocks, we use 3×3 kernels with 1× 1 strides and 1× 1 paddings. Channels are denoted below block names in Suppl. Fig. 11. For the pooling operations, we employ 2× 2 kernels for average pooling in both pool 1, pool 2, and pool 3. The LSTM is a single-layer LSTM. A linear layer is used for classification.

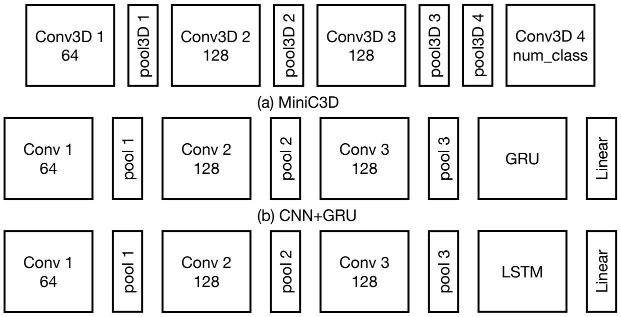
Comparison With Full Model. We show the classification results of UCF101 and HMDB51 on these models and full C3D in Suppl. Tab. 9. In Suppl. Tab. 9, we observe that (1) on MiniC3D, the classification accuracy of UCF101 is lower than that on full C3D; (2) on MiniC3D, the classification accuracy of HMDB51 can even exceed that on full C3D.

11.2. Details of Temporal Analysis

The temporal analysis experiments in Sec. 3.3 are conducted with DM [30] algorithm and CNN+GRU model as detailed before. 16 frames are sampled from each video with temporal stride 12, and we set the target synthetic video length also 16. For a fair comparison of time and space complexity, all the experiments are run on one NVIDIA V100 GPU (16GB), 8 cores of Intel Xeon 5218 CPU, and 20 GB memory. The learning rate for synthetic images is set to 1.0 and that for network updating is set to

ApplyEyeMakeup
BalanceBeam
GolfSwing
MTT
ApplyEyeMakeup
BalanceBeam
GolfSwing
MTT+Ours

Figure 10. Optical Flows of MTT for MiniUCF IPC=1.



(a) MiniC3D

(b) CNN+GRU

(c) CNN+LSTM

Figure 11. Structure of Models.

Dataset	IPC	lr_img	batch_real
MiniUCF	1	30	64
	5	100	64
HMDB51	1	30	64
	5	300	64
Kinetics400	1	100	64
	5	500	128
SSv2	1	10	64
	5	100	128

(a) DM [30]

Dataset	IPC	lr_dynamic	lr_hal	batch_real
MiniUCF	1	1e-4	1e-5	64
	5	1e3	1e-6	64
HMDB51	1	10	1e-6	64
	5	10	1e-5	64
Kinetics400	1	1	1e-5	64
	5	10	1e-5	128
SSv2	1	1	1e-5	64
	5	100	1e-5	128

(b) DM+Ours

Table 11. Hyper-parameters for DM and DM+Ours.

0.01. The models are trained for 10,000 iterations with a real batch size of 64, which is observed as enough for the training convergence in our experiments.

Dataset	IPC	lr_img
MiniUCF	1	1e-3
	5	1e-3
HMDB51	1	1e-3
	5	1e-3

(a) FRePo [32]

Dataset	IPC	lr_dynamic	lr_hal
MiniUCF	1	1e-4	1e-3
	5	1e-1	1e-3
HMDB51	1	1e-1	1e-3
	5	1e-2	1e-3

(b) FRePo+Ours

Table 12. Hyper-parameters for FRePo and FRePo+Ours.

11.3. Details of Full Experiments

In the experiments, we initially fine-tune the parameters for methods without our paradigm (naively adapted methods) to achieve the best possible results. To ensure a fair comparison, we strive to maintain consistency in all other parameter settings, making adjustments only to the learning rates associated with the unique dynamic information and $\mathcal{H}$ in the methods with our paradigm.

Structure of $\mathcal{H}$. We employ two different $\mathcal{H}$. The simple one only has one Conv3D block with 3×3× 3 kernels, 1× 1× 1 stride, and 1× 1× 1 padding, while the other one has one Conv3D block and one ConvTranspose3d block with middle channel 8. With the exception of DM+Ours for MiniUCF IPC=1, we consistently utilize the former.

Hyper-parameters for Distillation. We have thoroughly documented the parameters employed in our experiments in Suppl. Tab. 10 11 12. Parameters not explicitly mentioned default to the values specified in the original implementation code. The specific meanings of all mentioned parameters are detailed below:

lr_img: learning rate used to update distilled video.

lr_teacher: learning rate used to train expert trajectories of real videos and training trajectories of distilled videos. We set it to 0.01 by default.

batch_syn: number of distilled videos to match real videos at every iteration.

batch_real: number of real videos to be matched at every iteration.

syn_steps: steps of training trajectories of distilled videos to match expert trajectories at every iteration.

lr_dynamic: learning rate used to update dynamic memory.

lr_hal: learning rate used to update $\mathcal{H}$.

expert_epochs: steps of expert trajectories to be matched at every iteration. We set it to 1 by default.

max_start_epoch: the max starting step of expert trajectories to be matched. We set it to 10 by default.

We train 30 expert trajectories for MiniUCF and HMDB51 [11], and 20 for Kinetics400 [3] and Something-Something V2 [8]. Regarding whether to update lr_teacher during the training process, we retain relatively better results for each task, such as in the case of MTT for MiniUCF IPC=1, where the performance without updating lr_teacher surpasses that with updating.

Hyper-parameters for Evaluation. For evaluation on FRePo and FRePo+Ours, we set the learning rate to 1e-4 and trained for 1,000 epochs on MiniC3D. For other evaluations, we configure the learning rate to be 1e-2 and conduct training for 500 epochs on MiniC3D. In the case of the cross-architecture generalization test on CNN+GRU and CNN+LSTM, we set the learning rate to 1e-2 and trained for 100 epochs.

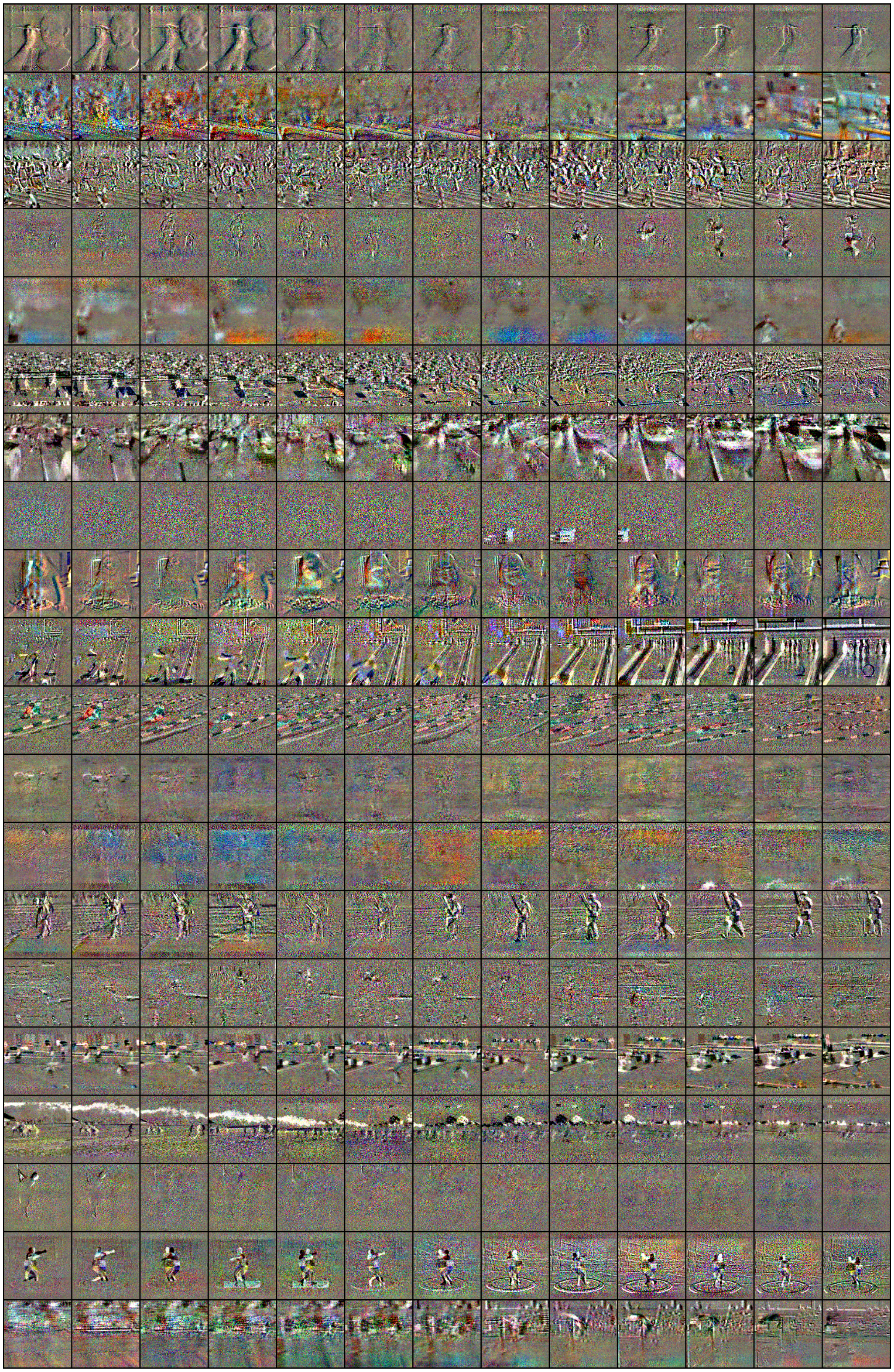


Figure 12. Inter-frame Differences of MTT for MiniUCF IPC=1.

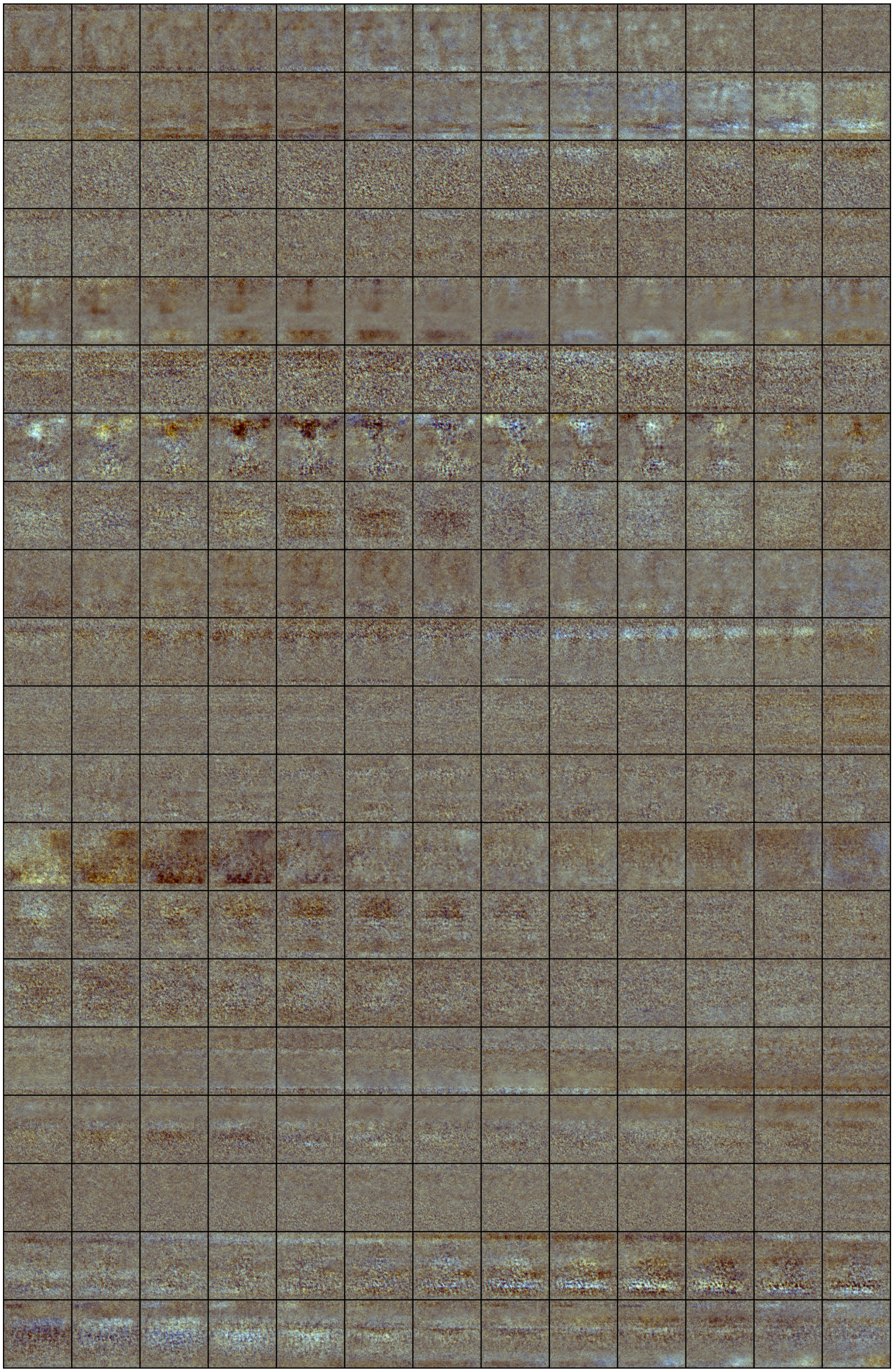


Figure 13. Inter-frame Differences of MTT+Ours for MiniUCF IPC=1.