6. Appendix

6.1. Evaluation Protocol

Our work is about self-supervised pretraining of video representations. For evaluation, we mostly perform transfer learning experiments following the standard linear evaluation protocol commonly used by recent self-supervised image representation learning approaches [8, 42]. Our video representation is first pretrained on unlabeled videos from a large pretraining dataset.

We then transfer the self-supervised representations to the target dataset, by training a linear classifier on top of the frozen representations. This linear classifier is trained on labeled examples from the training split of the target dataset. Accuracy on the test split of the target dataset is used to measure the representation quality. We list the pretrain and target datasets used to generate the results in our main submission in Table A1.

6.2. More Model Ablations

1. Number of transformer layers. We vary the number of transformer layers (1,2,4,8) for our projection head, which receives encoded time augmentations as additional input. Performance on SSv1 linear evaluation can be found in Table A2. We observe that the performance begins to slightly saturate at four layers.

No. Layers	Top-1 Acc.	Top-5 Acc.
1	26.9	56.2
2	30.0	60.3
4	31.2	61.4
8	31.3	62.4

Table A2: **Impact of number of layers in the transformer projection head** on Something-Something v1. Time shift encoding is used for all runs. The performance begins to gradually saturate at four layers. The transformer projection head is only applied during pre-training, and is not used in downstream tasks.

2. Number of Pretraining Epochs. We ablate the number of pretraining epochs when evaluated on SSv1, UCF101 and HMDB51. We observe in Table A3 that pretraining for more epochs helps improve representation quality, as also observed by [8], and it saturates at 500 epochs.

Epochs	SSv1	UCF101	HMDB51
200	29.8	71.4	43.6
500	32.2	84.3	53.6
800	33.1	83.6	53.0

Table A3: **Impact of number of training epochs** on SSv1, UCF101 and HMDB51, using linear eval on frozen features.

3. Results on SSv2. We follow the same setup as Table 2 and study the impact of crop and time encodings when both the pretraining and target datasets are SSv2. Results are shown in Table A4. We observe a similar trend as in SSv1: encoding time outperforms the no encoding baseline, and composing time and crop encodings further improves performance.

Enc. Crop	Enc. Time	Top-1 Acc.	Top-5 Acc.
✗	✗	40.0	72.4
✓	✗	40.1	72.4
✗	✓	42.3	74.5
✓	✓	**43.5**	**75.3**

Table A4: **Results on crop and time encodings on on SSv2** under a linear eval protocol. Trend is consistent with SSv1.

4. Types of Action Classification. In addition to results on SS, we also show results on standard action classification benchmarks UCF101 and HMDB51 under two settings - using all frames and using only the first frame in Table A5. We only show results with time encoding - we find that unlike SSv1 and SSv2, using the crop encoding hurts the performance. This is interesting and we conjecture that the benefit of augmentation encoding depends on the downstream task at hand: for fine-grained tasks that require some level of spatial reasoning (*e.g.* object localisation is needed to tell *picking up* from *putting down* in SSv1.), awareness of spatial augmentations is helpful; however for scene-level classification (*e.g.* UCF101 and HMDB51) it might be beneficial to be invariant to those augmentations.

Table A5 shows a similar trend for encoding time as that on SSv1, improving over the baseline. The relative improvement is bigger for first frame classification vs using all frames, however for both cases, the relative improvement is smaller than on SSv1. Finally, we also report results on the Kinetics-400 dataset: Without encoding time shifts, the linear evaluation top-1 accuracy is 55.3%. With encoding, the accuracy improves to 57.0%. The relative improvement is similar to that on UCF-101 and HMDB-51, and smaller than on Something-Something. These are consistent with previous observations [64] that temporal information is more important for the Something-Something dataset.

5. Nearest neighbor retrieval. We also validate our learned representations using the nearest neighbor retrieval benchmark. We follow the standard evaluation protocol [18, 32]: For each query video in the test set, we retrieve its top k nearest neighbors in the training set. A correct retrieval is deemed when any of the nearest neighbors belongs to the same category as the query video. We follow the linear evaluation procedure, and extract the visual representations from the visual encoders $f(\cot)$. For each video, we uniformly sample two windows of 32 frames and average

Table No.	Pretrain Data (Unlabeled)	Target Data (Train)	Target Data (Eval)
1,2,3,5	SSv1 train split	SSv1 train split	SSv1 val split
4,7	Kinetics-400 train split	UCF/HMDB train splits	UCF/HMDB val splits
6	SElse 'Base' train split	SElse 'Novel' train split	SElse 'Novel' val split

Table A1: Pretraining datasets for self-supervised representation learning with CATE, and target datasets for linear evaluation for results reported in the main paper.

Input	Encode time	UCF	HMDB
All frames	✗	83.01	52.77
All frames	✓	**84.32**	**53.57**
First frame	✗	73.67	38.69
First frame	✓	**75.50**	**40.13**

Table A5: **Effect of time encoding on UCF101 [52] and HMDB51 [31]** We show results for both early action classification (first frame) and regular action classification (all frames). We use frozen features: i.e. pretrained representations trained on Kinetics-400 are fixed and classified with a linear layer. Encoding time helps in both settings, albeit slightly more for early action classification.

their extracted representations. They are then L_2 normalized for retrieval. Following the standard protocol, we report results on the first split of UCF-101 and HMDB-51. As shown in Table A6 and A7, CATE significantly outperforms previous approaches in the video retrieval benchmark.

Method	top 1	top 5	top 10	top 20	top 50
OPN [32]	19.9	28.7	34.0	40.6	51.6
SpeedNet [5]	13.0	28.1	37.5	49.5	65.0
VCP [34]	19.9	33.7	42.0	50.5	64.4
Temporal SSL [25]	26.1	48.5	59.1	69.6	82.8
MemDPC† [18]	40.2	63.2	71.9	78.6	-
CATE	**54.9**	**68.3**	**75.1**	**82.3**	**89.9**

Table A6: Nearest neighbor retrieval evaluation on UCF-101 split 1. †: with Flow

Method	top 1	top 5	top 10	top 20	top 50
VCP [34]	6.7	21.3	32.7	49.2	73.3
MemDPC† [18]	15.6	37.6	52.0	65.3	-
CATE	**33.0**	**56.8**	**69.4**	**82.1**	**92.8**

Table A7: Nearest neighbor retrieval evaluation on HMDB-51 split 1. †: with Flow

6. Per-class breakdown on SSv1. Table A8 shows the classes that benefit the most and the least when crop augmentation is encoded by CATE. As discussed in the main paper, the trend is consistent with results for time encoding, and indicates that crop encoding leads to representation that better captures spatial information.

We further zoom into pairs of categories in Figure A1 with t-SNE plots. We extract representations from the test split of SSv1, where the representational model from the top row is learned by CATE with crop and time encoding, while the bottom row is learned without augmentation encoding. We pick categories that are sensitive of temporal ordering, such as *moving away* or *approaching something with camera*, or *pretending to put* or *show something* behind something. We observe that CATE in general leads to representation that better separates these fine-grained actions, where no encoding leads to data points from different categories (red and blue in the figure) mix with each other.

Label	ΔAP
Lifting something up completely, then letting it drop down	13.5
Pulling something from right to left	13.2
Moving something and something away from each other	13.2
Dropping something in front of something	12.6
Moving something down	12.2
Pretending to sprinkle air onto something	-7.0
Folding something	-8.6
Pretending or failing to wipe something off of something	-10.0
Moving away from something with your camera	-11.6

Table A8: Classes that benefit the most and the least with **crop encoding** on SSv1. We sort the classes by their differences on Average Precision.

6.3. Results on CLEVR and DSprites

Additionally, we further study the impact of crop encoding by using two image benchmarks that explicitly require spatial reasoning. The first dataset is CLEVR [28] with 70,000 training and 15,000 validation images. It is a diagnostic dataset which contains multiple objects of diverse shape and location configurations. We follow the setup used by [68] and evaluate on two tasks: **Count** which requires counting the total number of objects, and **Dist** which requires predicting the depth of the closest object to the camera, where the depth is bucketed into 6 bins. Both tasks are formulated as classification tasks. The second dataset is DSprites [36] which contains a single object floating around in an image, with various shape, scale, orientation and location. We use the **Location** task which requires predicting the (x, y) center location of the object. The x and y coordi-

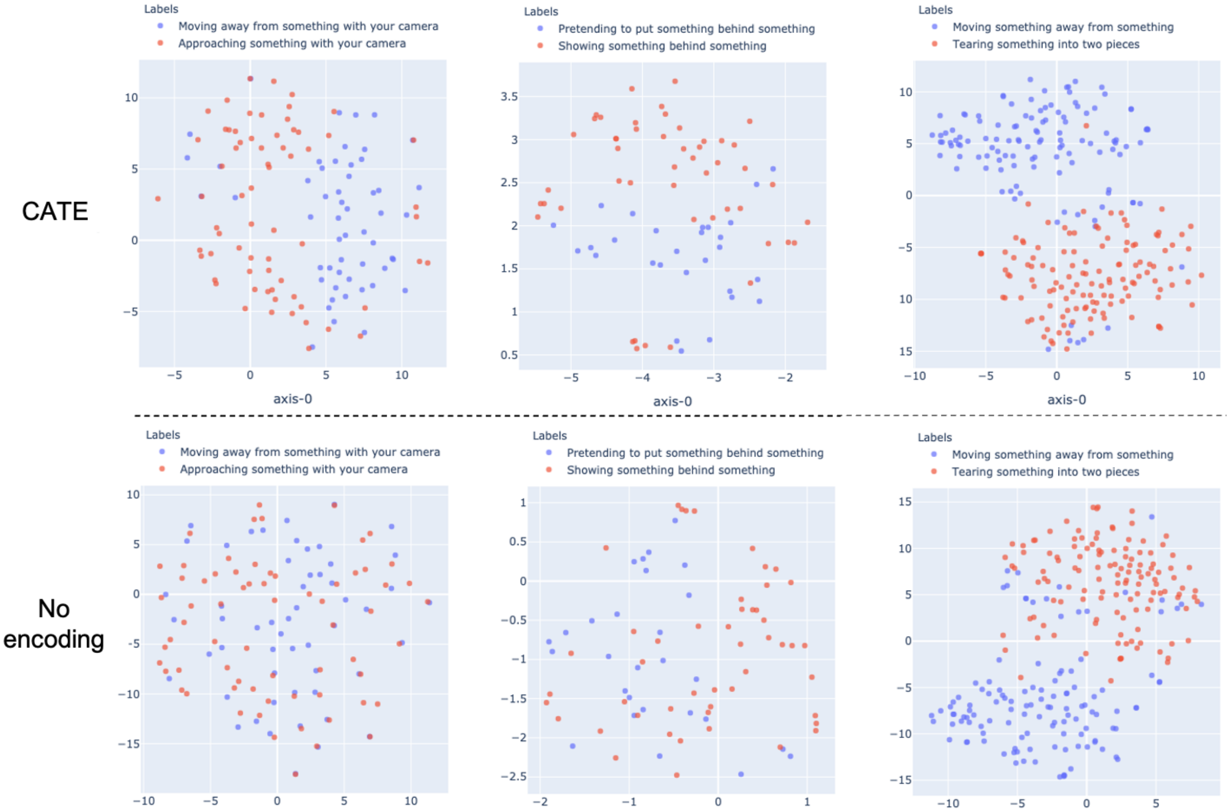

Figure A1. t-SNE plots computed on the test split of SSv1 videos. The top row uses representations learned by CATE with time and crop encoding, the bottom rows uses representations learned without any augmentation encodings. For each column, we zoom into two categories which are colour-coded using red and blue. We observe that CATE in general leads to representations that better separate fine-grained action categories which are sensitive to temporal information (*e.g. moving away or approaching something with camera*) Best viewed in colour.

nates are bucketed into 16 bins each. We report the geometric mean of classification accuracy on the bucketed x and y coordinates.

For both benchmarks, we train CATE using the same setups as we did with videos, except that the visual encoder is now a 2D ResNet-50, and the learning rate is reduced by 5x. We pretrain and evaluate on the datasets themselves.

Crop Enc.	CLEVR-Count [28]	CLEVR-Dist [28]	DSprites [36]
	65.3	64.3	28.1
✓	**68.8**	**66.9**	**38.8**

Table A9: Ablation of crop encoding on downstream tasks that require spatial reasoning, such as counting the number of objects, or localising objects in bucketed x, y coordinates.

The linear evaluation performance is shown in Table A9. We observe that encoding crop improves the transfer learning performance on all three tasks that require spatial reasoning, which further validates our conjecture.