Chirality in Action: Time-Aware Video Representation Learning by Latent Straightening

Piyush Bagad **Andrew Zisserman**
VGG, Dept. of Engineering Science, University of Oxford

Abstract

Our objective is to develop compact video representations that are sensitive to visual change over time. To measure such time-sensitivity, we introduce a new task: chiral action recognition, where one needs to distinguish between a pair of temporally opposite actions, such as "opening *vs*. closing a door", "approaching *vs*. moving away from something", "folding *vs*. unfolding paper", *etc*. Such actions (i) occur frequently in everyday life, (ii) require understanding of simple visual change over time (in object state, size, spatial position, count . . .), and (iii) are known to be poorly represented by many video embeddings. Our goal is to build time aware video representations which offer linear separability between these chiral pairs. To that end, we propose a self-supervised adaptation recipe to inject time-sensitivity into a sequence of frozen image features. Our model is based on an auto-encoder with a latent space with inductive bias inspired by *perceptual straightening*. We show that this results in a compact but time-sensitive video representation for the proposed task across three datasets: Something-Something, EPIC-Kitchens, and Charade. Our method (i) outperforms much larger video models pre-trained on large-scale video datasets, and (ii) leads to an improvement in classification performance on standard benchmarks when combined with these existing models.

1 Introduction

The ever-increasing scale of video content on the Internet demands efficient and compact descriptors that can be readily used for classification, ranking and search. The goodness of video descriptors (or video representations) is largely measured in terms of action recognition on standard benchmarks such as Kinetics-400 [11], UCF101 [73] and Something-Something [26] to name a few. While action recognition performance provides a reliable single measure for the representation, more insight is obtained by establishing how well a given video descriptor encodes various aspects of the video such as objects, scene context, motion and temporal dynamics. We can coarsely categorize these aspects into: *static* properties (objects, scene, *etc*) and *dynamic* properties (motion, visual change, *etc*). It is well established that, apart from Something-Something, most contemporary video benchmarks tend to focus more on static properties [33, 10, 42, 47, 106]. While there has been an effort to shift the focus to evaluating dynamic [57, 26, 71, 47], properties of actions are still entangled with static understanding without a clear definition of dynamics. In this work, our objective is to study a specific time-sensitive property: understanding how well video descriptors encode *visual change* in a video.

What do we mean by "understanding" visual change? Consider an example action pair: "a person climbing up a ladder" vs. "a person climbing down a ladder". In this case, the vertical position of the person changes over time and it is temporally opposite in the two actions. An ideal video descriptor should encode this change and use it distinguish between such action pairs. We call such action pairs*'chiral'*, and the task of distinguishing between them *'chiral action recognition'*. Where can we find such actions? These are quite common in everyday life, and humans effortlessly recognize them.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

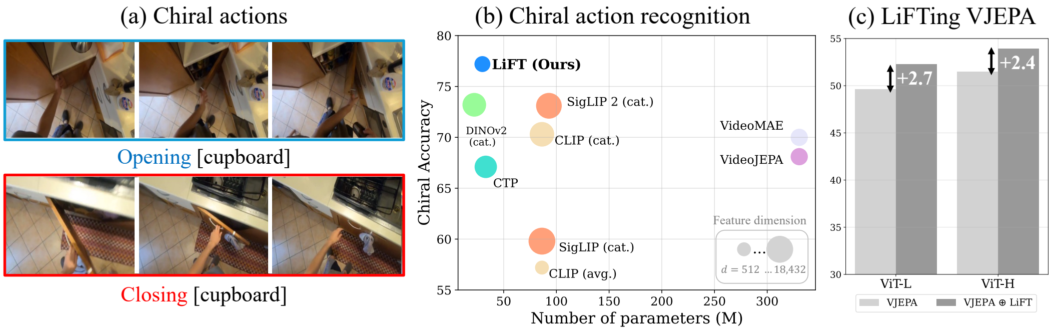


Figure 1: (a) We introduce *chiral actions*: temporally opposite action pairs to test time-awareness of video descriptors. We build a meta-dataset of chiral actions by mining SSv2, EPIC and Charades. To build time-aware descriptors, we propose LiFT (Linearized Feature Trajectories) that disentangles a sequence of DINOv2 features into a *static* and *dynamic* descriptor. (b) LiFT outperforms contemporary video and image models in recognizing chiral actions across the three datasets. Moreover, LiFT descriptors can be plugged into existing models (VideoJEPA or VideoMAE) to improve action recognition on standard benchmarks. (c) shows a linear probe of LiFT with VideoJEPA on SSv2.

In this work, we mine chiral pairs from three existing datasets (SSv2 [26], EPIC-Kitchens [16], and Charades [72], to set up a chiral evaluation benchmark.

Turning to the descriptors, most existing video representations are obtained by two classes of methods, either (i) self-supervised video embeddings – natively multi-frame models, trained on millions of videos [5, 77, 87], or (ii) per-frame image models adapted for video data that are usually trained for specific datasets [55, 62]. We borrow from both lines of work and propose a self-supervised adaptation recipe that yields general video descriptors that outperform much larger models [87, 5] for chiral action recognition. Specifically, we hypothesize that a representation will be time sensitive if the per-frame features form a smooth trajectory in latent space. Inspired by Perceptual Straightening [30], we operationalize this by learning a model that maps per frame features from a strong image model to ordered points on lines in latent space. We show that the two vectors representing these high-dimensional lines yield time-aware video descriptors. We call our model *LiFT* for Linearized Feature Trajectories. Qualitatively, we show that LiFT learns compact video descriptors that encode a smooth, continuous approximations of the feature trajectories. Quantitatively, we show that LiFT descriptors are time-aware: they can distinguish between chiral action pairs across three datasets without specialized fine-tuning.

While LiFT descriptors achieve strong results on chiral action recognition, can they be more generally useful, say by combining with other video models? In this spirit, we evaluate linear and attentive probes with LiFT descriptors combined with video models such as VideoJEPA [5] on four standard action recognition datasets: Kinetics-400 [11], UCF-101 [73], HMDB-51 [40] and SSv2 [26]. We show that the combination of LiFT and a given video model always outperforms solely using the video model, across different video models, across all four benchmarks. This demonstrates that the time-sensitivity in LiFT preserves information that is complementary to standard video models, which in turn helps *lift* performance on action recognition benchmarks. In summary, our contributions are:

1. We propose a new task called chiral action recognition which requires discounting static context and accounting for the dynamic change in a video. We formulate a meta-dataset from three action recognition datasets to benchmark this task.
2. We propose LiFT: a self-supervised recipe to adapt DINOv2 features into a compact, time-sensitive, and general video descriptor. LiFT outperforms much larger video models (e.g., 10× bigger and trained on over 6× samples) by over 7% on the proposed chiral benchmark.
3. We demonstrate that LiFT encodes time-sensitive information that is complementary to contemporary video models. We show that combining LiFT descriptors with video models such as VideoJEPA [5], VideoMAE [77] and InternVideo2.5 [87] lifts performance with linear as well as attentive probes as compared to only using these models.

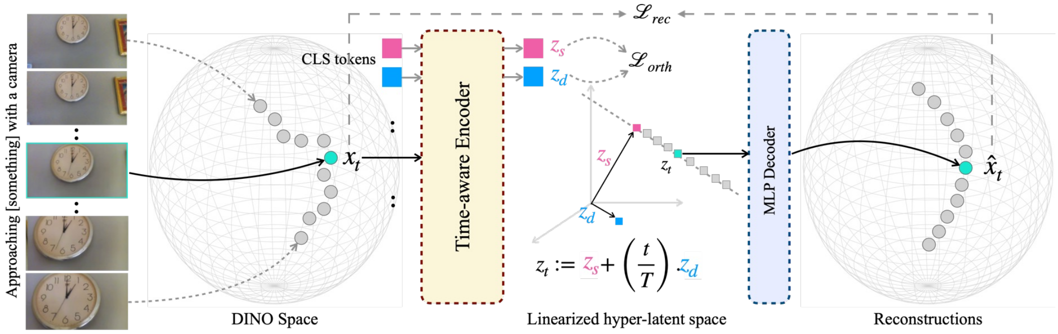


Figure 2: **Linearized Feature Trajectories (LiFT).** We propose LiFT as a simple adaptation of image features to obtain time-aware video descriptors. First, we encode each frame independently with a DINOv2 backbone. Then, we pass them through a Transformer encoder with two learnable tokens: $\mathbf{z}_s$ (static) and $\mathbf{z}_d$ (dynamic). Next, inspired by Perceptual Straightening [30], we enforce linearity in the latent space which enables reconstruction of the trajectory with only the two learnable tokens. We do not show position encodings and projection layers for brevity. The network is trained with the usual reconstruction loss and an orthogonality regularization between the static and dynamic tokens. Once the model is trained, an input video is represented by concatenation of $\mathbf{z}_s, \mathbf{z}_d$.

2 LiFT: Video Representation by Linearized Feature Trajectories

Our objective is to learn a single time-aware descriptor vector for a given video. We want to avoid training large video models [5, 77] from scratch given an academic compute budget. In such a scenario, usually Parameter-Efficient Fine-Tuning (PEFT) is employed to adapt image models for video data [55, 56]. However, PEFT is dataset-specific supervised tuning while our goal is to obtain a more general video descriptor while still adapting an image model without any label supervision.

Considering these desiderata, we propose a simple recipe based on adapting a strong image model with reconstruction in the latent space. Our central hypothesis is that for videos that depict a visual change, the per-frame features lie on smooth trajectories that encode such change over time. While these trajectories tend to be non-linear, we can map them to a latent space in which they are parametrized by a line. This is loosely inspired by the Perceptual Straightening Hypothesis [30]: humans perceive image sequences that are non-linear in pixel space as straight lines in the perceptual space. Thus, a video can be represented simply by the vectors that define this line in the latent space. Owing to this linear formulation, we call the model *LiFT*: Linearized Feature Trajectories.

The schematic diagram of LiFT is shown in Fig. 2. Formally, consider a video as a sequence of images $\{I_t\}_{t=1}^{T}, I_t \in \mathbb{R}^{C \times H \times W}, \forall t$. First, we encode each frame independently using an image model Φ . We only retain the global CLS token per frame.

$$\mathbf{x}_t = \Phi(I_t) \in \mathbb{R}^D, \forall t. \tag{1}$$

Then, we train an autoencoder network to reconstruct $\{\mathbf{x}_t\}_1^T$ while learning meaningful video descriptors in its latent space. Now, we describe the Encoder-Decoder network and how we train it.

Encoder. The encoder takes in sequence of frame features and outputs a descriptor for the sequence. First, we project the feature sequence to a potentially lower-dimensional space.

$$\mathbf{e}_t = P_{\downarrow}(\mathbf{x}_t) \in \mathbb{R}^d, \forall t. \tag{2}$$

Sinusoidal position encoding is added to encode the frame index. Then, a Transformer Encoder takes in this sequence $\{\mathbf{e}_t\}_1^T$ along with two learnable CLS-like tokens: $\mathbf{e}_s$ and $\mathbf{e}_d$ that are used to encode *static* and *dynamic* information respectively in the feature sequence.

$$\mathbf{z}_s, \mathbf{z}_d = \text{TransformerEncoder}\left(\mathbf{e}_s, \mathbf{e}_d, \{\mathbf{e}_t\}\right) \tag{3}$$

We collect these tokens output by the Transformer, denoted by $\mathbf{z}_s$ and $\mathbf{z}_d \in \mathbb{R}^d$. The overall video descriptor is given by their concatenation $\mathbf{z} \in \mathbb{R}^{2d}$.

Decoder. The decoder takes in $\mathbf{z}_s, \mathbf{z}_d$ and time index t and outputs feature vector as time t. In the latent space, we enforce a linearity constraint defined by $\mathbf{z}_s, \mathbf{z}_d$ as shown in Fig. 2.

$$\mathbf{z}_t := \mathbf{z}_s + \left(\frac{t}{T}\right).\mathbf{z}_d \in \mathbb{R}^d, \forall t \tag{4}$$

The decoder is a two-layer MLP that takes in $\mathbf{z}_t$ and outputs the reconstructed feature at time t.

$$\hat{\mathbf{x}}_t = \text{MLPDecoder}(\text{concat}([\mathbf{z}_s, \mathbf{z}_d])) \in \mathbb{R}^D, \forall t. \tag{5}$$

Training objective. We train the network with the usual reconstruction loss and a regularizer that encourages orthogonality between the static and dynamic latent vectors.

$$\mathcal{L} := \mathcal{L}_{\text{rec}} + \lambda \mathcal{L}_{\text{orth}} = \sum_{t=1}^{T} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2^2 + \lambda.\, \text{cos-sim}\left(\frac{\mathbf{z}_s}{\|\mathbf{z}_s\|_2}, \frac{\mathbf{z}_d}{\|\mathbf{z}_d\|_2}\right) \tag{6}$$

Note that this is unsupervised. Once trained, we discard the Decoder and use the Encoder to get a LiFT video descriptor $(\mathbf{z}_s, \mathbf{z}_d)$.

Time-awaremess of LiFT descriptors. To gain an insight into what LiFT learns, first, we visualize the joint tSNE embeddings of the original and reconstructed trajectories on sample videos in Fig. 3(a). We also visualize the tSNE embeddings of an example action pair in Fig. 3(b). These illustrate that: (a) LiFT outputs a smooth, continuous approximation of the original trajectories evident from the tSNE plot. In a sense, LiFT captures the "*arc of change*" depcited in the video; and, (b) Thanks to the simple linearization, LiFT is compelled to learn compact descriptors that can distinguish between temporally opposite actions such as "opening vs closing a door".

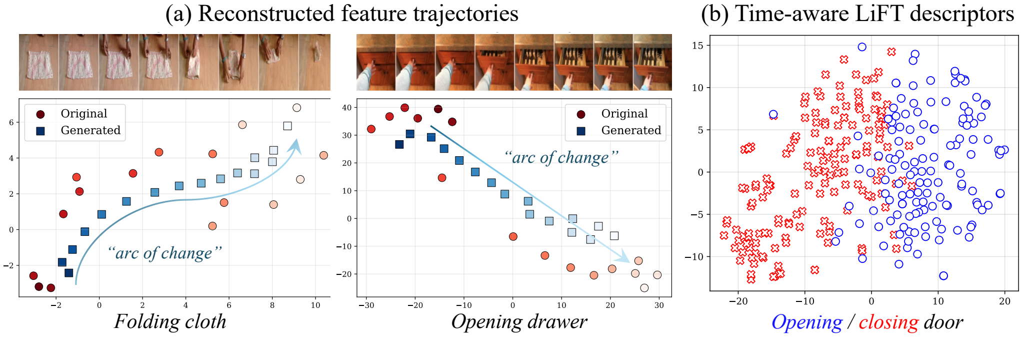


Figure 3: **Qualitative analysis.** (a) LiFT reconstructs a smooth, continuous approximation of the original feature trajectories roughly encoding the arc of visual change. (b) LiFT descriptors are time-aware: LiFT distinguishes between temporally opposite actions such as "opening/ closing door".

Implementation details. We use DINOv2 (ViT-S/14) [54] with registers [17] as the base image feature extractor with dimension D=384. We only use the CLS token output for each frame. We linearly sample T=16 frames from each video and compute the features ahead of training. The input feature sequence is first projected to the space of the Encoder $\mathbb{R}^D \rightarrow \mathbb{R}^d$; we choose d=384. The Encoder is a standard Transformer [14] with 4 layers and 8 attention heads each. The Encoder uses sinusoidal position encoding [14] to encode the frame index. The outputs from the two CLS tokens, $\mathbf{z}_s, \mathbf{z}_d \in \mathbb{R}^d$ are then projected to a space where we impose the linearity constraint: $\mathbb{R}^d \rightarrow \mathbb{R}^d$. The Decoder is an MLP with 2 hidden layers each followed by a GeLU activation [32] and LayerNorm [2]. Overall, the model has 8.7M trainable parameters beyond the 22M parameters of the frozen DINOv2 encoder. We provide more details on the architecture in the Supplemental. We also provide ablations varying d and the amount of training data in the Supplemental.

Training. We train LiFT on Kinetics-400 [11] which has about 240K videos. Since the image encoder is frozen, we pre-compute features which makes the training very efficient. The model is trained for 500 epochs with a batch size of 128 with Adam optimizer [39] with a learning rate of 0.001 and a LRPlateau scheduler.

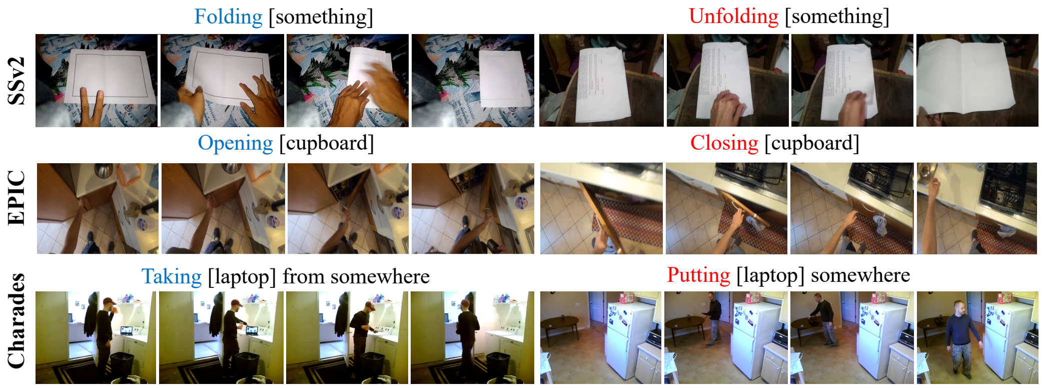


Figure 4: **Example chiral pairs** from each of the three datasets. To distinguish the actions in each pair, one needs to discount the spatial context and account for *what* is changing over time and *how*.

3 Chirality in Action (CiA) Dataset

To quantitatively measure time-awareness of LiFT (or other video descriptors), we propose a new task, namely, *chiral action recognition*. The study of time has covered aspects such as arrow of time [58, 89], order of frames [51, 100], recognizing temporally fine-grained actions [71, 102], or space-time tracking [94]. Prior work has developed specialized benchmarks and models for such tasks. In contrast, we want to measure time-awareness of a general video descriptor in recognizing simple everyday actions. It is well-established that existing action recognition benchmarks are biased to spatial understanding [33, 70]. Thus, we narrow down our focus on what we call *chiral actions*.

Chiral actions. In daily life, we often perform actions such as "closing/opening a door", "folding/unfolding a cloth" or "getting in/out of a car". Our proposal is that a good video model should distinguish between such temporally opposite actions. Loosely inspired by the notion of *visual chirality* [48], we call such action pairs as *chiral*, *i.e.*, pairs that are approximately mirror reflections along the time dimension. Consider the examples shown in Fig. 4. In distinguishing between these actions, one needs to discount the spatial context and account for the visual change over time.

Note that unlike in the study of arrow of time [89], we do not artificially reverse time-arrow but in a sense, our chiral actions have a naturally opposite arrow of time. The notion of chiral actions is also closely related to *reversible* actions studied in [60]. However, our work is complementary since we can use the methods in [60] to identify chiral actions and then evaluate video models on them. Finally, chiral actions, as we define them, are similar to *nearly symmetric actions* introduced in concurrent work by Ponbagavathi and Roitberg [59]. However, we build a meta-dataset of a more general mix of datasets that includes a richer set of actions, has both exo- and ego-centric videos and is larger in size.

Constructing CiA dataset. We build a meta-dataset out of chiral subsets of popular action recognition datasets. We identify three datasets to build a benchmark for chiral action recognition: Something-Something (SSv2) [26], EPIC-Kitchens (EPIC) [16], and Charades [72]. These datasets come with action recognition labels with separate verb and noun annotations. For each dataset, we build chiral pairs from the provided labels as follows.

1. We pass the list of action verbs to ChatGPT and ask it to find antonym pairs. We manually verify the output to remove pairs that have hallucinated verbs or those that are not visually antonymous.
2. For each verb pair, we group similar nouns together. For example, for verb pair "opening" vs "closing", nouns such as "door/cupboard/drawer" that represent visually similar actions are grouped. This group is represented by the triplet ("opening", "closing", "[door]"). Likewise, ("opening", "closing", "[box]") represents a separate chiral group where objects such as "tiffin/box/parcel", *etc* are grouped together. Thus, each chiral group is a triplet consisting of a pair of opposite verbs and the associated noun.
3. For each chiral group, we split the videos into train and test sets following the split defined in the original dataset.

Some basic numbers for each dataset are provided in Table 1 and visual examples are shown in Fig. 4.

Base dataset	Chiral groups	Avg videos/group	Example chiral group
Something-Something (SSv2) [26]	16	852.8	Folding / Unfolding [something]
EPIC-Kitchens (EPIC) [16]	66	412.2	Opening / Closing [door]
Charades [72]	28	768.4	Taking / Putting a [laptop]

Table 1: **Numbers for CiA meta-dataset.** We mine chiral action pairs in three existing action recognition datasets to build our benchmark. Visual examples are shown in Fig. 4.

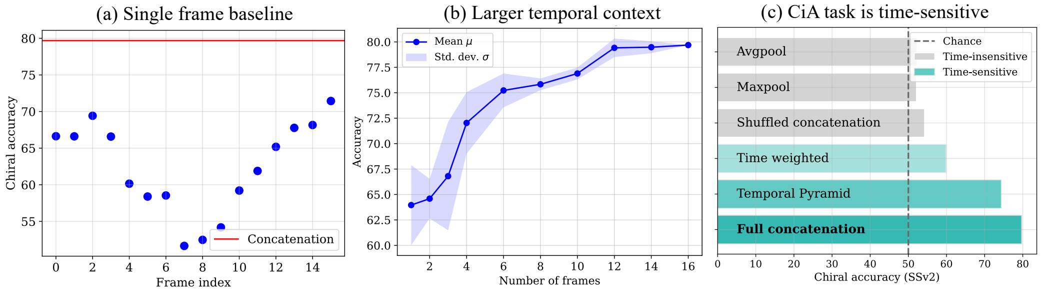


Figure 5: **Time-sensitivity of chiral action recognition.** On SSv2 (Chiral), we run simple baselines with DINOv2 features to test how time-sensitive the task is. In (a), (b), the video descriptor is obtained by concatenating per frame features. (a) Single frames chosen at the ends tend to do well but lag far behind using all frames. (b) Increasing number of frames provides consistent benefit while saturating around 16 frames. (c) Time insensitive pooling (*e.g.*, mean) of features is noticeably worse than time-sensitive pooling (*e.g.*, full concatenation or time-weighting).

Evaluation protocol. We measure time-sensitivity as a test of linear separability of chiral actions. For a given video model and a chiral group, we extract video representations for each video from the two antonym classes. We train a linear classifier in the feature space on the train set. We repeat this for each chiral group and report the average accuracy on test set across all groups. There are several reasons to choose this evaluation protocol: (i) We want the evaluation method to be as simple as possible so that the true strength of the representation is measured without confounding with the strength of the evaluation model (*i.e.*, in this case, a linear probe); (ii) Since there can be several chiral groups (e.g., K=16 in SSv2), it is computationally convenient to train simple linear probes rather than a more complicated and compute-heavy model for each group; (iii) Evaluating frozen features is much more common and practical [5, 12] as models get larger and evaluations need to be faster.

Properties of chiral action recognition. To analyze the time-sensitivity of chiral action recognition, we consider simple pooling baselines for DINOv2 features. The simplicity of the model enables us to study the task while discounting model strength. We show the results on SSv2 in Fig. 5. This shows that (a) A single frame chosen at either ends of the video yields a decent baseline as the end frames depict either the start or the end state of an action which is sometimes informative. However, the performance of such single frame baseline significantly lags using more frames as shown in (b). Increasing the temporal context with more frames consistently improves performance. (c) Finally, we compare time-insensitive pooling methods (*e.g.*, average) with time-sensitive ones (*e.g.*, full concatenation). We find that time-sensitive pooling does significantly better. Overall, this analysis highlights that the task at hand (chiral action recognition) benefits from time-aware ordering of many frames which establishes that it does not suffer as much from a single- or static-frame bias [33, 70].

4 Experiments

In this section, we first present results on our proposed chiral action recognition task. Then, we explore more general action recognition tasks where our descriptor is useful. In particular, in Section 4.1, we show that LiFT outperforms much larger video models in distinguishing between temporally opposite actions without specialized fine-tuning; and in Section 4.2, we show that plugging the LiFT descriptor with other general video models lifts their performance on standard action recognition benchmarks.

4.1 Chiral action recognition

Experimental details. We follow the evaluation protocol detailed in Section 3. For each chiral group, we compute video descriptors as described in Section 2 and train a linear classifier. We compare against two sets of baselines. (i) image models such as the image encoder in CLIP. (ii) Video models trained with self- or language supervision on large-scale video datasets. Typically, we sample $T{=}16$ frames linearly from the video and compute a single descriptor. For image models, we concatenate the per-frame features to obtain the video descriptor. For video models based on R(2+1)D architecture (e.g., TCLR [18]), we sample $T' = 8$ clips over the span of the video and concatenate clip features to obtain a single descriptor. For Transformer-based video models, if the model uses a CLS token, we treat that as descriptor or average pool all the output tokens unless stated otherwise.

Main result. From the results shown in Table 2, we observe that the proposed LiFT features achieve the best performance on SSv2, EPIC and Charade, while being compact ($d = 768$). Notably, LiFT beats much heavier video models such as VideoMAE, VideoJEPA and InternVideo2.5. Interestingly, naively concatenating image features with a strong model (DINOv2, SigLIP2) generally performs better or at par with much heavier video models (e.g., VideoMAE) reinforcing that the complete sequence of image features does retain rich information for chiral action recognition. However, naturally, naively concatenating yields very bulky descriptors which may be impractical, say, in indexing a database with millions of videos. Less surprisingly, Transformer-based video models outperform ResNet (R(2+1)D) based models.

Finally, we find that careful feature pooling can make a notable difference. For example, InternVideo2.5/VideoMAE/VideoJEPA output a sequence of tokens per frame. Average pooling over space and concatenating over time does much better than average pooling over space and time. Nevertheless, average pooling output tokens (e.g., from VideoMAE) is still reasonable and time sensitive because (i) it does not have an explicit CLS token as a single video embedding, (ii) it comprises of 3D space-time tokens enhanced by temporal position encoding; different frames interact with each other in every layer of the transformer.

				Chiral Accuracy ↑		
Model	**Architecture**	**Pooling**	**D ↓**	**SSv2**	**EPIC**	**Charades**
Chance	-	-	-	50.0	50.0	50.0
Image models with naive concatenation						
CLIP [63]	ViT-B/16	Average	512	53.5	63.4	54.7
CLIP [63]	ViT-B/16	Concat.	8192	71.6	71.6	67.7
BLIP2 [43]	ViT-B/16 + Q-Former	Concat.	12288	73.2	70.3	67.3
SigLIP [101]	ViT-B/16	Concat.	18432	57.9	66.2	55.2
SigLIP 2 [78]	ViT-B/16	Concat.	12288	76.8	74.7	67.8
DINOv2 [54]	ViT-S/14	Concat.	6144	79.7	74.1	65.8
Image models adapted for video (trained on Kinetics-400)						
ST-Adapter [55]	ViT-B/16	Learned	768	50.5	63.7	54.4
DiST [62]	ViT-B/16	Learned	512	52.1	59.9	55.9
Video models						
Tubelet Contrast [75]	R(2+1)D	Concat.	4096	64.6	62.8	58.9
TCLR [18]	R(2+1)D	Concat.	4096	67.9	62.5	58.8
CTP [84]	R(2+1)D	Concat.	4096	78.8	64.4	58.0
VideoMAE [77]	ViT-L/16x16x2	Average	1024	80.3	70.5	59.1
VideoMAEv2 [85]	ViT-B/16x16x2	Average	768	65.3	67.5	55.5
SIGMA [67]	ViT-B/16x16x2	Average	768	66.5	69.1	56.1
MME [74]	ViT-B/16x16x2	Average	768	78.4	70.8	57.5
VideoJEPA [5]	ViT-L/16x16x2	Average	1024	80.4	67.4	56.4
InternVideo 2.5 [87]	InternViT-6B	Average	4096	55.8	66.1	55.4
VideoMAE [77]	ViT-L/16x16x2	Time concat.	8192	85.7	75.0	66.1
VideoJEPA [5]	ViT-L/16x16x2	Time concat.	8192	85.4	70.8	57.1
InternVideo 2.5 [87]	InternViT-6B	Time concat.	32768	80.0	70.9	62.8
LiFT (Ours)	ViT-S/14	Learned	768	**86.6**	**75.5**	**69.5**

Table 2: **Results on chiral action recognition.** (1) Our method (LiFT) of efficiently adapting sequential information in DINOv2 features has the best performance on the chiral splits of all three datasets. (2) On average, sequence of image features contain stronger discriminative information for chiral actions in comparison to native video models.

Ablation study. We run ablation over key design choices such as the image encoder in LiFT. Results of ablation study on SSv2 are shown in Table 3. We note that self-supervised image features such as iBOT [105] and DINOv2 [54] outperform language-supervised models such as CLIP [63] or SigLIP2 [78]. We hypothesize that since DINOv2/iBOT are better at capturing spatial details, their feature trajectories of a video tend to better capture smooth visual change compared to language-supervised models. From the last two rows, we also establish that using the orthogonal loss does provide a small benefit in chiral action recognition.

What kinds of change are easier to understand? We categorize the visual change involved in each chiral action pair. For example, in "moving towards/away from the camera", the object size or depth changes or in "taking/putting one of many objects on table", the object count changes. We average the performance across all chiral pairs that depict a given kind of visual change across all three datasets and report in Table 4. We find that LiFT features shine in distinguishing chiral actions that involve change in object state or count but struggle in those with change in position along x-axis.

Image encoder	Architecture	SSv2 (Chiral)
CLIP	ViT-B/16	75.9
SigLIP2	ViT-B/16	77.8
BLIP2	ViT-B/16 + Q-Former	75.7
iBOT	ViT-B/16	80.3
DINOv2	ViT-B/14	85.4
DINOv2	ViT-L/14	85.9
LiFT w/o $\mathcal{L}_{\text{orth}}$	ViT-S/14	85.9
LiFT	ViT-S/14	**86.6**

Table 3: **Ablation on image encoders.** (i) self-supervised image features (*e.g.*, iBOT/DINOv2) outperform language-supervised features (*e.g.*, CLIP), (ii) with DINOv2, features out of larger models do not necessarily show improvement, (iii) using orthogonality loss helps by better disentangling $\mathbf{z}_s, \mathbf{z}_d$.

Change type	VMAE	VJEPA	LiFT
Dist. bet. objects	70.8	87.5	87.5
Object count	64.2	62.4	72.4
Object size/depth	96.8	96.8	96.8
Object state	72.9	66.3	80.7
Spatial position ↔	96.3	96.1	75.2
Spatial position ↕	91.5	89.7	93.6

Table 4: **Performance across kinds of change.** Color green denotes performance at par or better than competing models and red denotes worse. LiFT features shine in distinguishing chiral actions that involve change in object state or count but struggle in those with change in position along x-axis.

4.2 LiFTing video models on standard benchmarks

While LiFT outperforms much heavier video models in recognizing chiral actions, can it help improve performance on standard action recognition? We conduct an extensive linear probe evaluation across four standard datasets: Kinetics-400 (K400) [83], UCF-101 [73], HMDB-51 [40] and SSv2 [26]. The experimental details are provided in the Supplemental. As shown in Table 5a, while LiFT by itself does not beat top video models, concatenating LiFT with such video models consistently lifts their performance. This indicates that LiFT descriptors have complementary information. Furthermore, in Table 5b, we ablate over kinds of probes and model sizes used. We consistently observe a benefit with LiFT. Interestingly, with VideoJEPA as well as VideoMAE, ViT-L combined with LiFT even outperforms a scaled up ViT-H. Thus, overall, an adapter such as LiFT when combined with standard video models can provide strong video representations useful for classification, retrieval and search.

5 Related Work

Human perception of videos and time. Psychologists have tried to understand how humans perceive visual change in videos (*e.g.*, motion) for a long time [88, 81]. More recently, Hénaff et al. [30] present a remarkable finding: visual system in humans and macaques transforms complex pixel dynamics in videos into straighter temporal trajectories [30, 31]. Straighter trajectories make predictions easier and predictions are a fundamental part of human perception. Inspired by this insight, we learn an auto-encoder on image feature trajectories with linearity baked in the latent space. Although there is some prior work inspired by Perceptual Straightening [25, 53, 29], we apply it to a sequence of image features and show that it leads to more general time-aware representations.

Time-aware video representations. Temporally pooling image sequences has been a classical way of representing videos. Carefully crafted pooling in pixel space [8], in motion/flow space [9] and in embedding spaces [21] have been devised. Since the prominence of deep learning on videos, time has been creatively used as a source of self-supervision: space-time jigsaw [38], time arrow [89, 60], time order [95, 92, 24], speed [6], tracking [34, 83],contrasting temporal views [61, 18, 66],

Model	K400	UCF	HMDB	SSv2
Chance	0.25	0.99	1.96	0.58
LiFT	55.4	86.6	65.2	30.8
VJEPA	59.8†	91.3	76.1	49.6†
VJEPA $\oplus$ LiFT	63.7	92.6	78.0	52.3
Δ	+3.9	+1.3	+1.9	+2.7
VideoMAE	55.0	83.6	66.5	38.3
VideoMAE $\oplus$ LiFT	63.6	88.8	72.6	46.3
Δ	+8.6	+5.2	+6.1	+6.0
InternVid2.5	62.8	88.2	71.9	23.4
InternVid2.5 $\oplus$ LiFT	65.9	90.3	75.3	35.9
Δ	+3.1	+2.1	+3.4	+11.5

(a) LiFT combined with video models lifts their performance results across four action recognition benchmarks.

Arch.	Probe	Base	Base$\oplus$LiFT	Δ
VideoJEPA				
ViT-L	Non-lin.	51.7	54.2	+2.5
ViT-L	Attentive	65.9†	66.9	+1.0
ViT-L	Linear	49.6†	52.3	+2.7
ViT-H	Linear	51.5	53.9	+2.4
VideoMAE				
ViT-L	Non-lin.	43.9	50.1	+6.2
ViT-L	Attentive	61.5	63.7	+2.2
ViT-S	Linear	19.4	37.3	+17
ViT-B	Linear	25.6	41.1	+16
ViT-L	Linear	38.3	46.3	+6.0
ViT-H	Linear	40.0	46.9	+6.9

(b) LiFT consistently improves performance of video models across model scale and probes.

Table 5: **Results on standard action recognition datasets.** LiFT improves probing accuracies with standard video models across datasets and model sizes. †Note: The numbers for VideoJEPA are obtained with our experimental setup. We could not precisely reproduce the numbers reported in the paper [5] even using their codebase. We have reached out to the authors for clarification.

cycle consistency in time [20] or explicitly modeling temporal dynamics [103, 35, 15]. Modern video encoders are based on Transformers [7, 1, 49, 77, 85, 5]. Data-efficiency [75, 77] and time-sensitivity [67, 76, 96, 19] of video models continue to be active areas of research [69]. In this work, we investigate time-sensitivity of existing models through chiral action recognition and propose a simple recipe to embed videos based on summarizing trajectories of image features. Concurrent to our work, Xue et al. [93] propose a reinforcement learning-based training strategy to instill arrow of time awareness in video LLMs. Our chiral actions are related to the arrow of time, but we do not artificially reverse the arrow of time in a video; instead we aim to distinguish actions that are naturally opposite along the arrow of time.

Action recognition benchmarks. Early datasets for action recognition in videos include UCF-101 [73], HMDB-51 [40], Sports-1M [36] and Kinetics [11]. Transformers [82] prompted the rise of multimodal video datasets with text [50, 4], audio [23, 13] and 3D [79, 27]. LLMs led to the rise of instruction-tuning datasets [44, 104] and benchmarks [45, 22, 57] for videos. However, the community has repeatedly discovered that a majority of these do not actually test for time; a single frame or an unordered set of frames would suffice to recognize the action in the video [33, 10, 42, 47, 106]. SSv2 [26], Diving-48 [47] introduced temporally sensitive actions while other datasets evaluate specific aspects: causal/counterfactual reasoning [97, 91, 57], compositionality [28, 99], concept-binding [37, 68], temporal prepositions [3] and verbs [52, 70]. In this work, we propose chiral action recognition that evaluates video features in discriminating temporally opposite actions. Our definition of chirality is related to that of equivariant actions in Price and Damen [60] but their aim was more to discover actions invariant/equivariant to time flipping. Chirality is also related to *nearly symmetric actions* in concurrent work by Ponbagavathi and Roitberg [59]. However, unlike [59], we propose a more general video embedding model trained in an unsupervised manner.

Efficient adaptation of image models to videos. Given the computational cost of training video models from scratch, Parameter Efficient Fine-Tuning (PEFT) methods to adapt image models for videos have emerged [55, 65, 80, 62, 46]. Since we use frozen DINOv2 features, our work also adapts image model for video recognition. However, PEFT methods are usually trained separately for each downstream dataset and generally used in a supervised learning setup. Our method is more generally applicable. It is trained in an unsupervised manner on Kinetics-400 and the resulting video embeddings are shown to be applicable for chiral action recognition across three datasets.

6 Discussion and Conclusion

In an effort to develop time-sensitive video descriptors, we proposed Linearized Feature Trajectories (LiFT): a simple recipe to adapt DINO per-frame features with an auto-encoder with an inductive bias inspired by Perceptual Straightening [30]. As a measure of time-sensitivity, we introduce chiral action recognition to distinguish between temporally opposite actions such as “opening vs. closing a door". We created the CiA meta-dataset with chiral pairs mined from three public datasets: SSv2 [26], EPIC [16], and Charades [72]. On CiA, we show that LiFT outperforms much heavier video models including VideoJEPA [5] and VideoMAE [77] while being compact. Furthermore, we show that the time-sensitive LiFT descriptors contain information that is complementary to standard video models. For example, LiFT when combined with VideoJEPA lifts performance across four action recognition benchmarks: Kinetics [11], UCF [73], HMDB [40] and SSv2 [26].

Future work. Since we only use per frame CLS tokens, LiFT likely misses out on some spatial details, especially horizontal translation as shown in Table 4. Investigating ways of mitigating this, *e.g.*, using a sequence of dense feature maps, is an open avenue for future research. Furthermore, since our recipe is self-supervised, combining it with other compute-heavy self-supervised pre-training paradigms such as Masked Modeling [77, 5] or Autoregression [64, 90] should be interesting avenues to imbue more time-sensitivity into these representations.

Acknowledgments. This research is funded by the EPSRC Programme Grant VisualAI EP/T028572/1, and a Royal Society Research Professorship RSRP\R\241003. We thank Ashish Thandavan for support with infrastructure, and Sindhu Hegde and Makarand Tapaswi for useful discussions. We also thank the anonymous reviewers that helped improve this work.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A Video Vision Transformer. In *International Conference on Computer Vision (ICCV)*, 2021. 9

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer Normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[3] Piyush Bagad, Makarand Tapaswi, and Cees GM Snoek. Test of Time: Instilling Video-Language Models with a Sense of Time. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 9

[4] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in Time: A Joint Video and Image Encoder for End-to-end Retrieval. In *International Conference on Computer Vision (ICCV)*, 2021. 9

[5] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting Feature Prediction for Learning Visual Representations from Video. *Transactions on Machine Learning Research*, 2024. 2, 3, 6, 7, 9, 10, 20, 21

[6] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the Speediness in Videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 8

[7] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is Space-Time Attention All You Need for Video Understanding? In *International Conference on Machine Learning (ICML)*, 2021. 9

[8] Hakan Bilen, Basura Fernando, Efstratios Gavves, Andrea Vedaldi, and Stephen Gould. Dynamic Image Networks for Action Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 8

[9] Aaron F. Bobick and James W. Davis. The Recognition of Human Movement Using Temporal Templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2001. 8

[10] Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the" Video" in Video-Language Understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 9

[11] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 4, 9, 10

[12] João Carreira, Dilara Gokay, Michael King, Chuhan Zhang, Ignacio Rocco, Aravindh Mahendran, Thomas Albert Keck, Joseph Heyward, Skanda Koppula, Etienne Pot, et al. Scaling 4D Representations. *arXiv preprint arXiv:2412.15212*, 2024. 6

[13] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A Large-scale Audio-visual Dataset. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020. 9

[14] Junlin Chen, Chengcheng Xu, Yangfan Xu, Jian Yang, Jun Li, and Zhiping Shi. Flatten: Video Action Recognition is an Image Classification Task. *arXiv preprint arXiv:2408.09220*, 2024. 4

[15] Siyi Chen, Minkyu Choi, Zesen Zhao, Kuan Han, Qing Qu, and Zhongming Liu. Unfolding Videos Dynamics Via Taylor Expansion. *arXiv preprint arXiv:2409.02371*, 2024. 9

[16] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling Egocentric Vision: The EPIC-Kitchens Dataset. In *European Conference on Computer Vision (ECCV)*, 2018. 2, 5, 6, 10

[17] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision Transformers Need Registers. *arXiv preprint arXiv:2309.16588*, 2023. 4

[18] Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. TCLR: Temporal Contrastive Learning for Video Representation. *Computer Vision and Image Understanding*, 2022. 7, 8, 20

[19] Ishan Rajendrakumar Dave, Mamshad Nayeem Rizve, Chen Chen, and Mubarak Shah. Timebalance: Temporally-invariant and Temporally-distinctive Video Representations for Semi-supervised Action Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 9

[20] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal Cycle-Consistency Learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 9

[21] Basura Fernando, Efstratios Gavves, José Oramas, Amir Ghodrati, and Tinne Tuytelaars. Rank Pooling for Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2016. 8

[22] Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-MME: The First-Ever Comprehensive Evaluation Benchmark of Multi-modal LLMs in Video Analysis. *arXiv preprint arXiv:2405.21075*, 2024. 9

[23] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio Set: An Ontology and Human-Labeled Dataset for Audio Events. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. 9

[24] Amir Ghodrati, Efstratios Gavves, and Cees GM Snoek. Video Time: Properties, Encoders and Evaluation. *arXiv preprint arXiv:1807.06980*, 2018. 8

[25] Ross Goroshin, Michael F Mathieu, and Yann LeCun. Learning to Linearize Under Uncertainty. *Advances in neural information processing systems*, 2015. 8

[26] 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 *International Conference on Computer Vision (ICCV)*, 2017. 1, 2, 5, 6, 8, 9, 10

[27] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the World in 3,000 Hours of Egocentric Video. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 9

[28] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. Agqa: A Benchmark for Compositional Spatio-temporal Reasoning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 9

[29] Anne Harrington, Vasha DuTell, Ayush Tewari, Mark Hamilton, Simon Stent, Ruth Rosenholtz, and William T Freeman. Exploring Perceptual Straightness in Learned Visual Representations. In *International Conference on Learning Representations (ICLR)*, 2022. 8

[30] Olivier J Hénaff, Robbe LT Goris, and Eero P Simoncelli. Perceptual Straightening of Natural Videos. *Nature neuroscience*, 2019. 2, 3, 8, 9

[31] Olivier J Hénaff, Yoon Bai, Julie A Charlton, Ian Nauhaus, Eero P Simoncelli, and Robbe LT Goris. Primary Visual Cortex Straightens Natural Video Trajectories. *Nature communications*, 2021. 8

[32] Dan Hendrycks and Kevin Gimpel. Gaussian Error Linear Units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016. 4

[33] De-An Huang, Vignesh Ramanathan, Dhruv Mahajan, Lorenzo Torresani, Manohar Paluri, Li Fei-Fei, and Juan Carlos Niebles. What Makes a Video a Video: Analyzing Temporal Information in Video Understanding Models and Datasets. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 5, 6, 9, 22

[34] Allan Jabri, Andrew Owens, and Alexei Efros. Space-Time Correspondence as a Contrastive Random Walk. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 8

[35] Dinesh Jayaraman and Kristen Grauman. Slow and Steady Feature Analysis: Higher Order Temporal Coherence in Video. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 9

[36] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale Video Classification with Convolutional Neural Networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 9

[37] Zeeshan Khan, CV Jawahar, and Makarand Tapaswi. Grounded Video Situation Recognition. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 9

[38] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised Video Representation Learning with Space-Time Cubic Puzzles. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2019. 8, 20

[39] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[40] 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 (ICCV)*, 2011. 2, 8, 9, 10

[41] Hyeongmin Lee, Jin-Young Kim, Kyungjune Baek, Jihwan Kim, Hyojun Go, Seongsu Ha, Seokjin Han, Jiho Jang, Raehyuk Jung, Daewoo Kim, et al. TWLV-I: Analysis and Insights from Holistic Evaluation on Video Foundation Models. *arXiv preprint arXiv:2408.11318*, 2024. 20

[42] Jie Lei, Tamara L Berg, and Mohit Bansal. Revealing Single Frame Bias for Video-and-Language Learning. *arXiv:2206.03428*, 2022. 1, 9

[43] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *International Conference on Machine Learning (ICML)*, 2023. 7

[44] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric Video Understanding. *arXiv preprint arXiv:2305.06355*, 2023. 9

[45] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A Comprehensive Multi-modal Video Understanding Benchmark. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 9

[46] Xinhao Li, Yuhan Zhu, and Limin Wang. Zeroi2v: Zero-cost Adaptation of Pre-trained Transformers from Image to Video. In *European Conference on Computer Vision (ECCV)*, 2024. 9

[47] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards Action Recognition Without Representation Bias. In *European Conference on Computer Vision (ECCV)*, 2018. 1, 9

[48] Zhiqiu Lin, Jin Sun, Abe Davis, and Noah Snavely. Visual Chirality. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5

[49] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video Swin Transformer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 9

[50] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *International Conference on Computer Vision (ICCV)*, 2019. 9

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

[52] Liliane Momeni, Mathilde Caron, Arsha Nagrani, Andrew Zisserman, and Cordelia Schmid. Verbs in Action: Improving Verb Understanding in Video-Language Models. In *International Conference on Computer Vision (ICCV)*, 2023. 9

[53] Julie Xueyan Niu, Cristina Savin, and Eero Simoncelli. Learning Predictable and Robust Neural Representations by Straightening Image Sequences. *Advances in Neural Information Processing Systems*, 2024. 8

[54] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning Robust Visual Features Without Supervision. *arXiv preprint arXiv:2304.07193*, 2023. 4, 7, 8

[55] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. ST-Adapter: Parameter-Efficient Image-to-Video Transfer Learning. *Advances in Neural Information Processing Systems*, 2022. 2, 3, 7, 9

[56] Jungin Park, Jiyoung Lee, and Kwanghoon Sohn. Dual-path Adaptation from Image to Video Transformers. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[57] Viorica Patraucean, Lucas Smaira, Ankush Gupta, Adria Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Mateusz Malinowski, Yi Yang, Carl Doersch, et al. Perception Test: A Diagnostic Benchmark for Multimodal Video Models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 1, 9

[58] Lyndsey C Pickup, Zheng Pan, Donglai Wei, YiChang Shih, Changshui Zhang, Andrew Zisserman, Bernhard Scholkopf, and William T Freeman. Seeing the Arrow of Time. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 5

[59] Thinesh Thiyakesan Ponbagavathi and Alina Roitberg. Order Matters: On Parameter-Efficient Image-to-Video Probing for Recognizing Nearly Symmetric Actions. *arXiv preprint arXiv:2503.24298*, 2025. 5, 9

[60] Will Price and Dima Damen. Retro-Actions: Learning'close'by Time-reversing'open'videos. In *International Conference on Computer Vision Workshops (ICCVW)*, 2019. 5, 8, 9

[61] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal Contrastive Video Representation Learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 8

[62] Zhiwu Qing, Shiwei Zhang, Ziyuan Huang, Yingya Zhang, Changxin Gao, Deli Zhao, and Nong Sang. Disentangling Spatial and Temporal Learning for Efficient Image-to-Video Transfer Learning. In *International Conference on Computer Vision (ICCV)*, 2023. 2, 7, 9

[63] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models from Natural Language Supervision. In *International Conference on Machine Learning (ICML)*, 2021. 7, 8

[64] Jathushan Rajasegaran, Ilija Radosavovic, Rahul Ravishankar, Yossi Gandelsman, Christoph Feichtenhofer, and Jitendra Malik. An Empirical Study of Autoregressive Pre-training from Videos. *arXiv preprint arXiv:2501.05453*, 2025. 10

[65] Hanoona Rasheed, Muhammad Uzair Khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Fine-tuned CLIP Models are Efficient Video Learners. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 9

[66] Adria Recasens, Pauline Luc, Jean-Baptiste Alayrac, Luyu Wang, Florian Strub, Corentin Tallec, Mateusz Malinowski, Viorica Pătrăucean, Florent Altché, Michal Valko, et al. Broaden Your Views for Self-supervised Video Learning. In *International Conference on Computer Vision (ICCV)*, 2021. 8

[67] Mohammadreza Salehi, Michael Dorkenwald, Fida Mohammad Thoker, Efstratios Gavves, Cees GM Snoek, and Yuki M Asano. SIGMA: Sinkhorn-Guided Masked Video Modeling. In *European Conference on Computer Vision (ECCV)*, 2024. 7, 9

[68] Darshana Saravanan, Darshan Singh, Varun Gupta, Zeeshan Khan, Vineet Gandhi, and Makarand Tapaswi. VELOCITI: Can Video-Language Models Bind Semantic Concepts Through Time? *arXiv preprint arXiv:2406.10889*, 2024. 9

[69] Madeline C Schiappa, Yogesh S Rawat, and Mubarak Shah. Self-supervised Learning for Videos: A Survey. *ACM Computing Surveys*, 2023. 9

[70] Laura Sevilla-Lara, Shengxin Zha, Zhicheng Yan, Vedanuj Goswami, Matt Feiszli, and Lorenzo Torresani. Only Time Can Tell: Discovering Temporal Data for Temporal Modeling. In *Winter Conference on Applications of Computer Vision (WACV)*, 2021. 5, 6, 9, 22

[71] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A Hierarchical Video Dataset for Fine-grained Action Understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 5

[72] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 5, 6, 10

[73] K Soomro. UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild. *arXiv:1212.0402*, 2012. 1, 2, 8, 9, 10

[74] Xinyu Sun, Peihao Chen, Liangwei Chen, Changhao Li, Thomas H Li, Mingkui Tan, and Chuang Gan. Masked Motion Encoding for Self-Supervised Video Representation Learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 7

[75] Fida Mohammad Thoker, Hazel Doughty, and Cees GM Snoek. Tubelet-contrastive Self-supervision for Video-efficient Generalization. In *International Conference on Computer Vision (ICCV)*, 2023. 7, 9, 22

[76] Fida Mohammad Thoker, Letian Jiang, Chen Zhao, and Bernard Ghanem. SMILE: Infusing Spatial and Motion Semantics in Masked Video Learning. *arXiv preprint arXiv:2504.00527*, 2025. 9

[77] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 3, 7, 9, 10, 20

[78] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual Vision-language Encoders with Improved Semantic Understanding, Localization, and Dense Features. *arXiv preprint arXiv:2502.14786*, 2025. 7, 8

[79] Vadim Tschernezki, Ahmad Darkhalil, Zhifan Zhu, David Fouhey, Iro Larina, Diane Larlus, Dima Damen, and Andrea Vedaldi. EPIC Fields: Marrying 3D Geometry and Video Understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 9

[80] Shuyuan Tu, Qi Dai, Zuxuan Wu, Zhi-Qi Cheng, Han Hu, and Yu-Gang Jiang. Implicit Temporal Modeling with Learnable Alignment for Video Recognition. In *International Conference on Computer Vision (ICCV)*, 2023. 9

[81] Jan PH Van Santen and George Sperling. Temporal Covariance Model of Human Motion Perception. *Journal of the Optical Society of America A*, 1984. 8

[82] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 9

[83] Shashanka Venkataramanan, Mamshad Nayeem Rizve, João Carreira, Yuki M Asano, and Yannis Avrithis. Is ImageNet Worth 1 Video? Learning Strong Image Encoders from 1 Long Unlabelled Video. *arXiv preprint arXiv:2310.08584*, 2023. 8

[84] Guangting Wang, Yizhou Zhou, Chong Luo, Wenxuan Xie, Wenjun Zeng, and Zhiwei Xiong. Unsupervised Visual Representation Learning by Tracking Patches in Video. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 7

[85] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. VideoMAE v2: Scaling Video Masked Autoencoders with Dual Masking. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 7, 9, 20

[86] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling Foundation Models for Multimodal Video Understanding. In *European Conference on Computer Vision (ECCV)*, 2024. 20

[87] Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, et al. InternVideo2. 5: Empowering Video MLLMs with Long and Rich Context Modeling. *arXiv preprint arXiv:2501.12386*, 2025. 2, 7, 20

[88] Andrew B Watson and Albert J Ahumada Jr. Model of Human Visual-motion Sensing. *Journal of the optical Society of America A*, 1985. 8

[89] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and Using the Arrow of Time. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5, 8

[90] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling Autoregressive Video Models. *arXiv preprint arXiv:1906.02634*, 2019. 10

[91] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next Phase of Question-Answering to Explaining Temporal Actions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 9

[92] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised Spatiotemporal Learning Via Video Clip Order Prediction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 8

[93] Zihui Xue, Mi Luo, and Kristen Grauman. Seeing the Arrow of Time in Large Multimodal Models. *arXiv preprint arXiv:2506.03340*, 2025. 9

[94] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning Spatio-temporal Transformer for Visual Tracking. In *International Conference on Computer Vision (ICCV)*, 2021. 5

[95] Charig Yang, Weidi Xie, and Andrew Zisserman. Made to Order: Discovering Monotonic Temporal Changes Via Self-supervised Video Ordering. *arXiv preprint arXiv:2404.16828*, 2024. 8

[96] Di Yang, Yaohui Wang, Quan Kong, Antitza Dantcheva, Lorenzo Garattoni, Gianpiero Francesca, and François Brémond. Self-supervised Video Representation Learning Via Latent Time Navigation. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2023. 9

[97] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. Clevrer: Collision Events for Video Representation and Reasoning. *arXiv preprint arXiv:1910.01442*, 2019. 9

[98] Xueyang Yu, Xinlei Chen, and Yossi Gandelsman. Learning Video Representations Without Natural Videos. *arXiv preprint arXiv:2410.24213*, 2024. 22

[99] Zhou Yu, Lixiang Zheng, Zhou Zhao, Fei Wu, Jianping Fan, Kui Ren, and Jun Yu. ANetQA: A Large-scale Benchmark for Fine-grained Compositional Reasoning Over Untrimmed Videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 9

[100] Sukmin Yun, Jaehyung Kim, Dongyoon Han, Hwanjun Song, Jung-Woo Ha, and Jinwoo Shin. Time is Matter: Temporal Self-supervision for Video Transformers. *arXiv preprint arXiv:2207.09067*, 2022. 5

[101] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In *International Conference on Computer Vision (ICCV)*, 2023. 7

[102] Chuhan Zhang, Ankush Gupta, and Andrew Zisserman. Temporal Query Networks for Fine-Grained Video Understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 5

[103] Heng Zhang, Daqing Liu, Qi Zheng, and Bing Su. Modeling Video as Stochastic Processes for Fine-grained Video Representation Learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 9

[104] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. LLaVA-Video: Video Instruction Tuning With Synthetic Data. *Transactions on Machine Learning Research*, 2025. 9

[105] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image BERT Pre-training with Online Tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 8

[106] Orr Zohar, Xiaohan Wang, Yann Dubois, Nikhil Mehta, Tong Xiao, Philippe Hansen-Estruch, Licheng Yu, Xiaofang Wang, Felix Juefei-Xu, Ning Zhang, et al. Apollo: An Exploration of Video Understanding in Large Multimodal Models. *arXiv preprint arXiv:2412.10360*, 2024. 1, 9

A Dataset: Chirality in Action (CiA)

Metadata and examples. In Table 6, we show the chiral groups constructed in SSv2. Similarly, we construct 66 chiral groups in EPIC and 28 groups in Charades. In Table 7, we show the number of videos in the chiral splits of each of the three datasets. We attach the chiral splits for all three datasets as part of the Supplemental. We also provide a single CSV file that includes the chiral groups for all three combined data sets. We also show more examples of chiral pairs from each of the three datasets in Fig. 6, Fig. 7 and Fig. 8. In general, since SSv2 has single canonical actions, it is a cleaner test bed for chiral action recognition. EPIC and Charades usually have a more cluttered visual context where cues for chiral recognition are more subtle.

Verb →	**Verb ←**	**Noun (object)**
Pulling [something] from left to right	Pulling [something] from right to left	[’something’]
Pushing [something] from left to right	Pushing [something] from right to left	[’something’]
Turning the camera left while filming [...]	Turning the camera right while filming [...]	[’something’]
Approaching [something] with your camera	Moving away from [something] with your camera	[’something’]
Closing [something]	Opening [something]	[’object’]
Closing [something]	Opening [something]	[’door’]
Closing [something]	Opening [something]	[’bottle’]
Closing [something]	Opening [something]	[’book’]
Closing [something]	Opening [something]	[’purse’]
Closing [something]	Opening [something]	[’drawer’]
Moving [...] and [...] away from each other	Moving [...] and [...] closer to each other	[’something’]
Moving [something] away from the camera	Moving [something] towards the camera	[’something’]
Moving [something] down	Moving [something] up	[’something’]
Putting [something similar to other things ...]	Taking [one of many similar things on the table]	[’something’]
Turning the camera downwards while filming [...]	Turning the camera upwards while filming [...]	[’something’]
Folding [something]	Unfolding [something]	[’something’]

Table 6: **Chiral groups in SSv2.** We construct 16 chiral groups in SSv2 by identifying temporally opposite verbs. Note that “opening vs. closing” is split across different objects representing entirely different actions. Noun “[‘something’]” denotes a placeholder which can include any appropriate object that fits with the action verb.

Dataset	**Chiral groups**	**Total videos**		**Avg. videos per chiral group**		**Avg. duration (s)**
		Train	*Validation*	*Train*	*Validation*	
SSv2	16	12216	1430	763.5	89.4	3.6
EPIC	66	24101	3108	365.1	47.1	1.6
Charades	28	16018	5498	572.1	196.4	8.6

Table 7: **CiA dataset size.** For each of the constituent datasets, we show the total number of videos in the proposed chiral split and also the average number of videos per chiral group. Note that we train one linear probe for each chiral group.

Time-sensitivity of CiA. In Fig. 9 and Fig. 10, we repeat the experiments to check time-sensitivity (Fig 5 in the main paper) of the CiA benchmark on all three datasets. Our inferences about time-sensitivity hold for all three datasets.

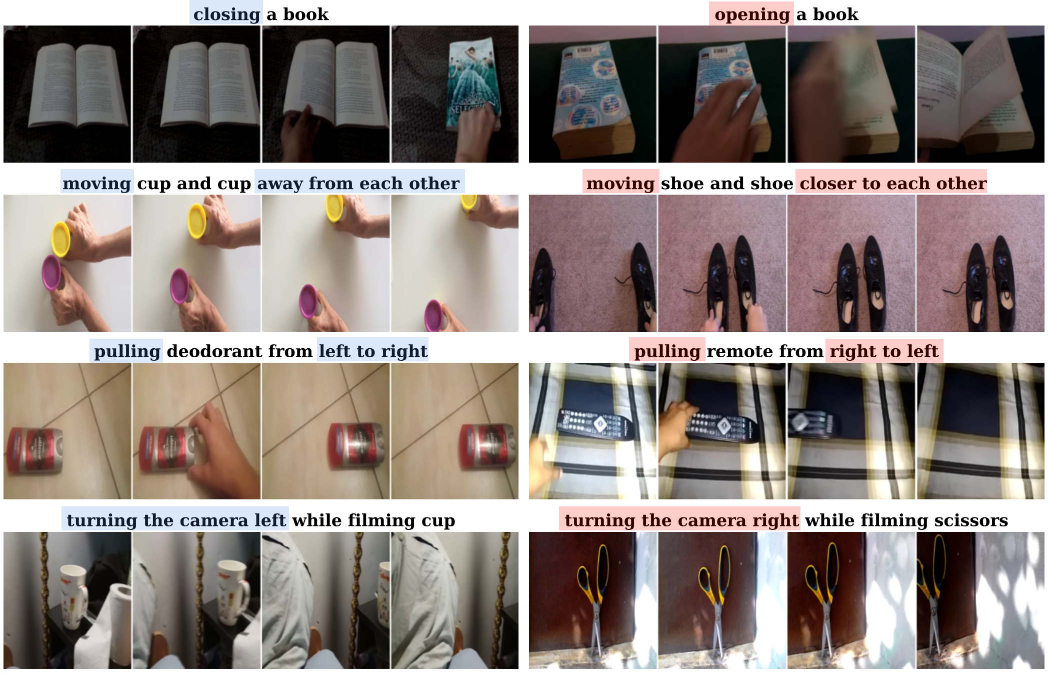

closing a book
opening a book
moving cup and cup away from each other
moving shoe and shoe closer to each other
pulling deodorant from left to right
pulling remote from right to left
turning the camera left while filming cup
turning the camera right while filming scissors

Figure 6: **CiA samples from SSv2.** More examples of chiral pairs from the train set of SSv2. The positive direction actions are marked in blue while the negative direction ones are marker in red.

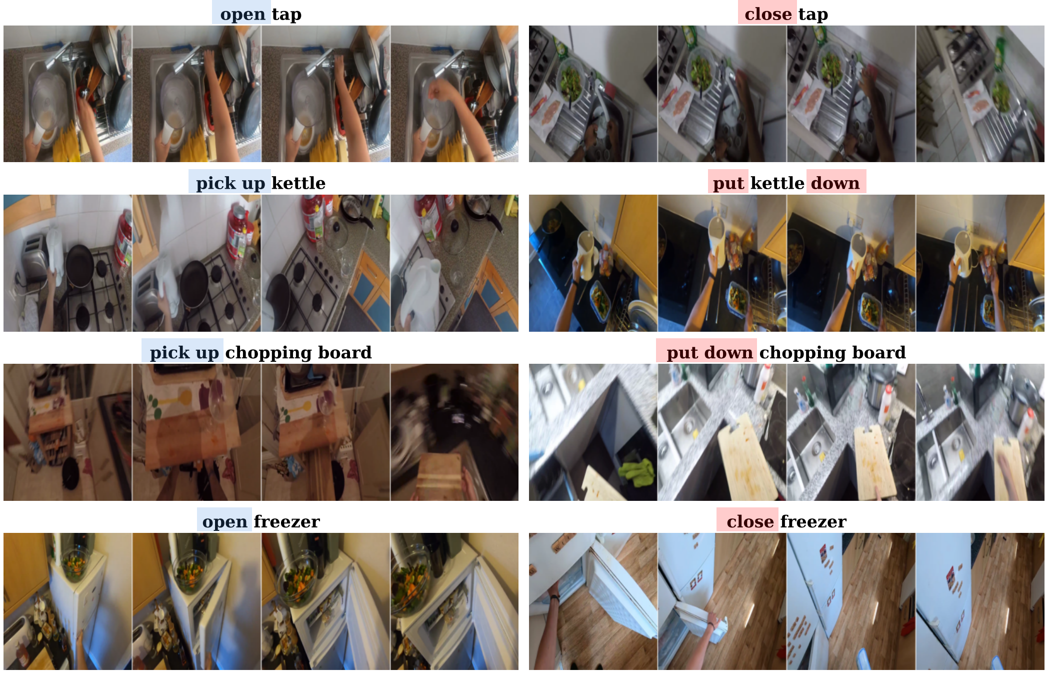


Figure 7: **CiA samples from EPIC.** More examples of chiral pairs from the train set of EPIC. The positive direction actions are marked in blue while the negative direction ones are marker in red.

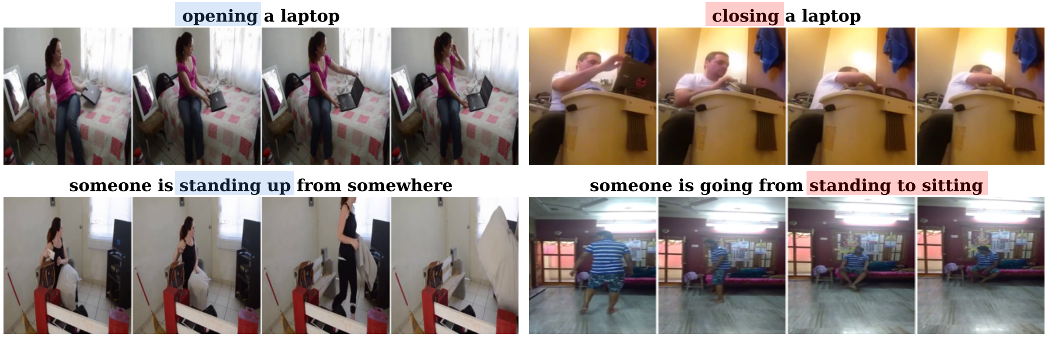

opening a laptop
closing a laptop
someone is standing up from somewhere
someone is going from standing to sitting

Figure 8: **CiA samples from Charades.** More examples of chiral pairs from the train set of Charades. The positive direction actions are marked in blue while the negative direction ones are marker in red.

B Model: Linearized Feature Trajectories (LiFT)

Architecture. A detailed sketch of the architecture is provided in Fig. 11. In LiFT, the Encoder takes in a sequence of features $\{\mathbf{x}_t\}_t$ and outputs two descriptor tokens $\mathbf{z}_s$, $\mathbf{z}_d$. First, a linear layer projection is applied $\mathbb{R}^D \rightarrow \mathbb{R}^d$. Then, Sinusoidal position encoding is added representing frame index t. The two CLS tokens $\mathbf{e}_s, \mathbf{e}_d$ are initialized randomly. Then, the CLS tokens along with the sequence tokens are passed through a Transformer with L=4 blocks and H=8 heads each. Each block has a multi-head self-attention (MHSA) layer followed by a FFN layer. Both the layers are preceded by LayerNorm layers. Then, the outputs for the two CLS tokens are projected with a linear layer ($\mathbb{R}^d \rightarrow \mathbb{R}^d$) followed by LayerNorm. This gives the latent descriptors $\mathbf{z}_s$ and $\mathbf{z}_d$.

The decoder takes in $\mathbf{z}_s, \mathbf{z}_d, t$ and outputs $\hat{\mathbf{x}}_t$. First, we construct an intermediate representation for the frame at index t using our linearity constraint in the latent space.

$$\mathbf{z}_t = \mathbf{z}_s + (t/T).\mathbf{z}_d \tag{7}$$

Then, this is passed to an MLP network with two hidden layers each followed by GeLU activation and LayerNorm. The first hidden layer maps $\mathbb{R}^d \rightarrow \mathbb{R}^{2d}$ and the second layer maps $\mathbb{R}^{2d} \rightarrow \mathbb{R}^{2d}$. This is followed by a linear projection ($\mathbb{R}^{2d} \rightarrow \mathbb{R}^D$) back to the DINOv2 space.

Compute resources. In order to train LiFT, we first compute and store feature vectors for DINOv2 ViT-S/14. This feature computation is run on 4 NVIDIA RTX A4000 16GB GPUs in parallel. It takes about 12 GPU hours to compute features for 250K videos in Kinetics-400. Once features are computed, LiFT is trained on a single consumer-grade GPU (*e.g.*, NVIDIA RTX A4000, Tesla P40, Quadro RTX 8000, NVIDIA RTX A6000). A single training run takes about 15 GPU hours.

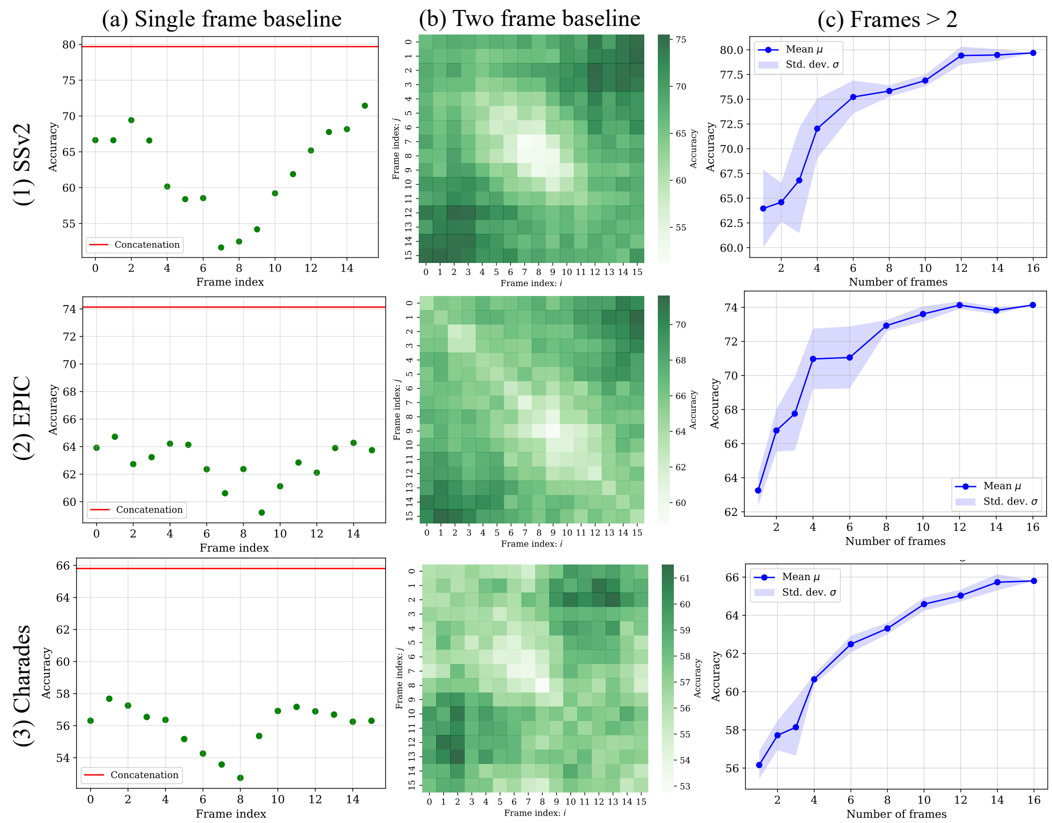


Figure 9: **Time-sensitivity of CiA: Part 1.** We repeat the experiment shown in Fig 5 (a)/(b) of the main paper for all datasets. Rows represent datasets while columns represent different properties of the task. (a) A single-frame baseline tends to do well on frames at the end of the video sequence since those usually encode either the start or end state of the action. (b) A two-frame baseline usually does best if the frames are picked at the two ends of the video. (c) As more frames are considered in the context, accuracy on chiral action recognition improves. Overall, these demonstrate that chiral action recognition is time-sensitive: it benefits predictably from more frames, especially at the ends.

C Experiments

C.1 Setup details

Details for chiral action recognition. To benchmark a given video model for chiral action recognition, we require a single descriptor vector for a video. There are two important details here: (i) *input processing pipeline*: Different methods differ in the way they sample frames, apply cropping operations, etc. (ii) *pooling*: existing methods [5, 77, 18] usually only represent short clips (sequence of frames with a fixed stride), so we need a way of pooling clip-level descriptors into a video-level descriptor. For (i), we follow the data pipeline for each model as provided. For (ii), depending on the method, we either average pool per-clip representations following [41] (*e.g.*, for VideoMAEv2 [85]) or concatenate them (*e.g.*, for 3D ResNet methods like TCLR [18]), or we hand-craft a pooling mechanism (*e.g.*, averaging spatial tokens for each frame and concatenating across time for InternVideo2.5 [86]). Investigating a general pooling method that gives more time-aware descriptors is an avenue we leave for future work. For image-based model, we sample T frames linearly and simply concatenate per-frame features to represent the video.

Details for standard action recognition. For the experiments with probing video models [38, 77, 87] with LiFT, we sample a single clip of T=16 frames with a stride of s=4, resize the short side and center crop to $(224, 224)$. Since VideoMAE, VideoJEPA and InternVideo2.5 all produce a sequence of space-time tokens without any global CLS token, we compute the average of all tokens to represent

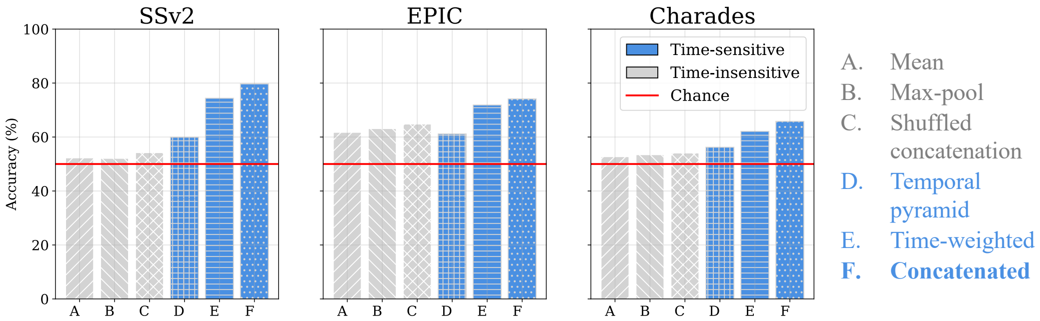


Figure 10: **Time-sensitivity of CiA: Part 2.** We repeat the experiment shown in Fig 5 (c) of the main paper across all three datasets. We show that time-insensitive pooling of per-frame features (*e.g*., average pooling) leads to much worse performance that with time-sensitive pooling (*e.g*., concatenation) on chiral action recognition. Note that all the pooling methods considered are non-parametric. This demonstrates the time-sensitivity of chiral action recognition since incorporating the time order of frames substantially improves performance.

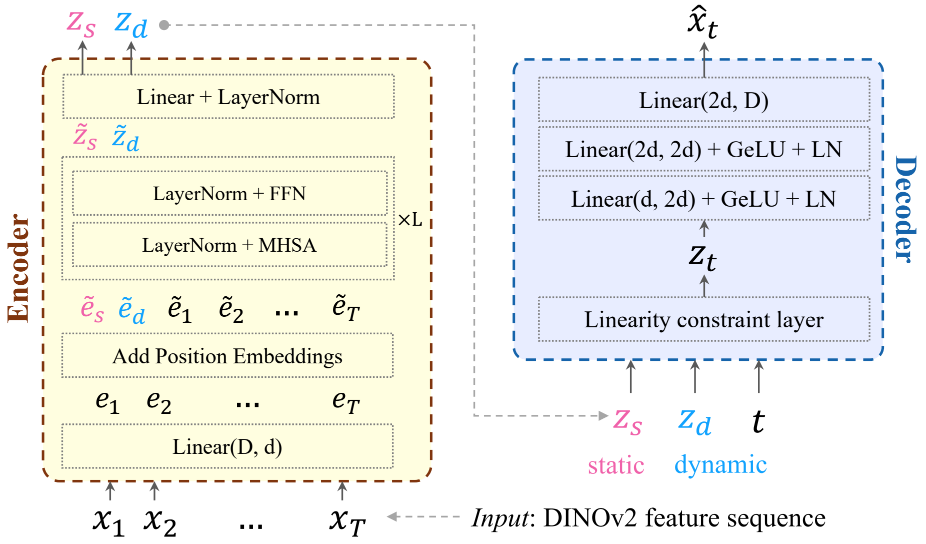


Figure 11: **LIFT architecture details.** The encoder takes in a sequence of DINOv2 features and outputs a video descriptor disentangled into static and dynamic vectors. The decoder reconstructs the feature sequence with linearity baked in the latent space.

the video in case of linear/non-linear probing. Then, we concatenate the LiFT descriptor with this descriptor and train a classifier head on top to output the action class. In case of *linear probe*, the classifier head is a linear layer. In case of *non-linear probing*, it is a two layer MLP with 512 hidden dimensions with ReLU non-linearity and Dropout of 0.1. In case of an attentive probe, following [5], we train a single attention layer with a learnable query to pool the space-time tokens into a single descriptor. Then, LiFT is concatenated with the query vector and a linear classifier layer is added on top of the concatenation. We train the probe for 100 epochs using Adam optimizer with learning rate of $1e^{-5}$ and LRPlateau scheduler.

C.2 Additional Ablations

Varying the latent dimension d. In Tab. 3(a), we vary the latent dimension of the Encoder in LiFT. While the number of parameters increases with d, we find that with $d = 384$, LiFT achieves the best performance while still being compact and containing only 8.7M parameters. Note that we do not account for the fixed DINOv2 parameters (22.1M) in this experiment.

Varying the amount of training data. In Tab. Appendix C.2 (b), we vary the amount of training data used to train LiFT with the unsupervised reconstruction loss. We fix the latent dimension to be $d = 384$ and note that the model capacity is fixed. For each row, we run the experiment with three different random seeds and report the average and standard deviation in accuracy. Surprisingly, even with 10% of the data, LiFT gets to 83.3% accuracy. With increasing data samples, the mean accuracy increases marginally. We hypothesize that this is due to two reasons. (i) The model size remains fixed (8.7M) and may not have the capacity to significantly benefit from more samples, (ii) since Kinetics-400 is known to be biased to static understanding (single frame or unordered set of frames [33, 70]), for videos with little visual change, reconstructing the per-frame feature trajectories may not have sufficient signal to inform LiFT. This experiment raises some interesting research questions. Is it possible to achieve time-sensitive video representations by training on selected, *temporally hard* samples only? Is using synthetic data with hand-crafted temporal patterns sufficient [75, 98]? We leave these questions for future work.

Latent dim d	**Accuracy**	**Parameters (M)**
192	85.9	2.3
256	85.8	4.0
384	**86.6**	8.7
512	84.9	15.3

(a) Varying the latent dimension d of Encoder.

Data frac.	**Accuracy**
0.1	83.3 ± 0.1
0.2	84.4 ± 0.2
0.4	85.9 ± 0.4
0.6	85.6 ± 0.6
0.8	85.8 ± 0.8
1.0	**86.2** ± 0.4

(b) Varying the % train data.

Table 8: **Ablations.** Both ablations are conducted on the chiral subset of SSv2. In (a) we vary the latent dimension of the LiFT encoder. We find the best performance with $d = 384$. In (b), we vary the amount of training data (Kinetics-400) used to adapt LiFT. The given % is uniformly randomly chosen from the entire dataset. Surprisingly, even with 10% of the data, LiFT gets to 83% accuracy. We hypothesize that at fixed model capacity, scaling up to more samples gives diminishing returns.

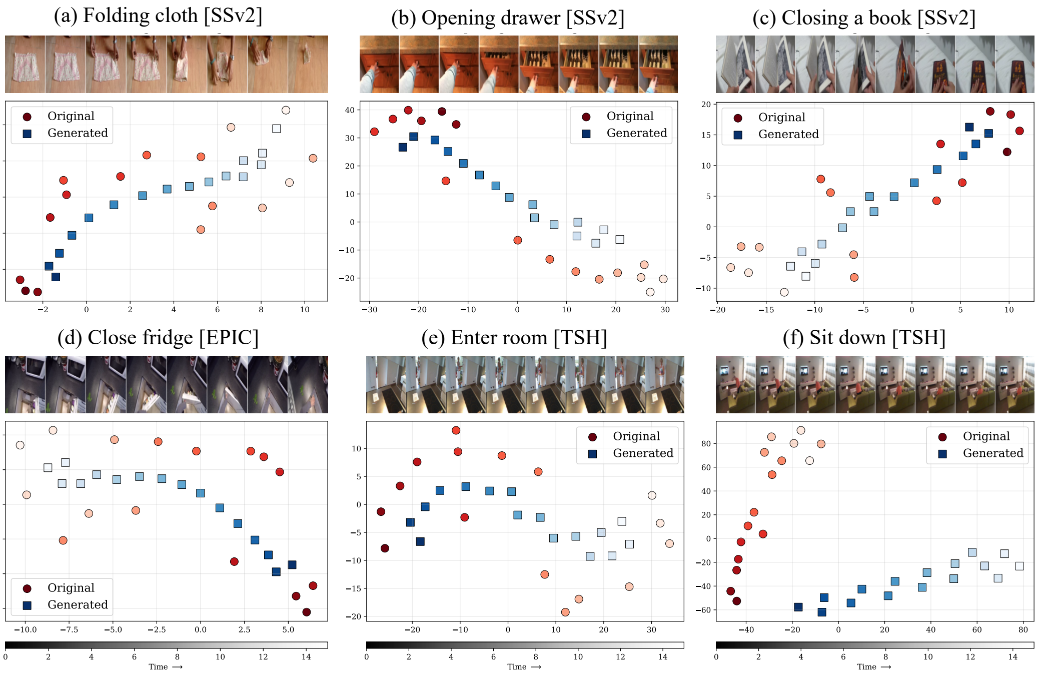
Error bars. To compute error bars, we train LiFT on Kinetics-400 with five different random seeds. The rest of the training configuration is kept constant across all runs. Then, we evaluate the trained models on our main task: chiral action recognition as described in the main paper across the three datasets, SSv2, Charades and EPIC-Kitchens. We report the mean and standard deviation in accuracy in Table 9. The table illustrates these results, highlighting the consistency of the model's performance.

Dataset	**Accuracy (%)**
SSv2	86.1 ± 0.3
EPIC	76.5 ± 0.8
Charades	70.3 ± 0.6

Table 9: **Error bars for LiFT.** Mean accuracy across five random seeds. LiFT remains fairly stable and the error bars emphasize the difference between LiFT and other video models.

C.3 Qualitative results

In Fig. 12, we show more examples with tSNE embeddings of the LiFT reconstructed feature trajectories. In most cases, LiFT reconstructs a smoother, continuous approximation of the original trajectory. Note that the original trajectory points seem more scattered than they actually are because tSNE optimizes for local neighborhood distances, which are dominated by closeness of points in the reconstructed trajectory. In case of Fig. 12(f), we observe a divergence between the true and reconstructed trajectories. In this case, the model likely fails to capture the (subtle) visual change which likely causes $\mathbf{z}_d$ to be inaccurate leading to the discrepancy.


(a) Folding cloth [SSv2]

(b) Opening drawer [SSv2]

(c) Closing a book [SSv2]

(d) Close fridge [EPIC]

(e) Enter room [TSH]

(f) Sit down [TSH]

Figure 12: **More qualitative samples of reconstructed features.** We show tSNE embeddings of original and reconstructed features for six videos. The red circles represent original features while blue squares represent reconstructions. Gradient of the color encodes the frame index (time). In general, LiFT tends to output a smooth, continuous approximation of the original feature trajectories in DINO space. (f) is an example failure case where the static token seems reasonable but the direction token is inaccurately predicted causing the direction of original and reconstructed trajectores to differ.

C.4 The curious case of horizontal motion

Based on Table 4, it seems that correctly encoding horizontal motion (distinguish between something moving → and →) is much harder than encoding vertical motion. Notably, we find that the base model itself (DINOv2-ViT-S/14) struggles with this kind of horizontal motion. Ideally, a change in horizontal spatial position ("moving left to right" vs "moving right to left") should result in dynamic tokens pointing in opposite directions. But this is conditioned on the base model reliably encoding the horizontal spatial position of an object at a given time. Our experiments in Table 10 confirm that the base model itself (DINOv2) does not accurately encode the horizontal spatial position of an object.

Change type	VideoMAE	VJEPA	DINOv2 (concat.)	LiFT
Distance between objects	70.8	87.5	83.3	**87.5**
Object count	64.2	62.4	69.5	**72.4**
Object size/depth	**96.8**	**96.8**	92.2	**96.8**
Object state	72.9	66.3	75.9	**80.7**
Spatial position ↔	**96.3**	96.1	75.7	75.2
Spatial position ↕	91.5	89.7	79.7	**93.6**
Average	82.1	83.1	79.4	**84.4**

Table 10: **LiFT is comparable or superior to much larger video models for all types of visual changes except horizontal shift.** On horizontal shift (e.g., "Pulling something from left to right vs. right to left"), LiFT is worse than these video models. As evident from the DINOv2 (concat.) column, we confirm that this is because the concatenated base DINOv2 features do not encode such motion as well as the video models.

Change type	**LiFT**	**(1.) w/ 224 → 448**	**(2.) w/ WebSSL**	**(3.) w/ TTR**
Distance between objects	87.5	**95.8**	91.7	91.7
Object count	72.4	73.7	**73.9**	73.2
Object size/depth	96.8	92.9	92.2	**96.9**
Object state	**80.7**	80.3	**80.7**	79.2
Spatial position $\leftrightarrow$	75.2	**82.4**	79.7	77.9
Spatial position $\updownarrow$	93.6	**94.4**	92.8	92.4
Average	84.4	**86.6**	85.2	85.2

Table 11: **We show three directions that improve the performance on encoding horizontal shift motion.** TTR denotes test-time rotation augmentation.

Furthermore, we dug deeper into analyzing the DINOv2 feature sequences for horizontal vs vertical shift samples. We hypothesize that the root cause of the difference in performance of DINOv2 concat. (and consequentially, LiFT) on horizontal vs vertical shifts is due to anisotropic sensitivity of DINO feature sequence, i.e., DINO features vary less with horizontal movement vs vertical movement. Below, we explain our experimental setup and the observations.

To have a perfectly controlled test setting, we generate N=2000 synthetic sequences with a checkerboard background and a colored disc that moves either horizontally (from left end of the image to right) or vertically (from top to bottom) at a constant rate. We compute the DINOv2 feature vector for each frame in the sequence. To measure the variation over time, we compute the variance over time and then average it across the feature dimensions. We call this Time Variance (TV). We compute the TV for each sequence and then average it over all sequences for horizontal (or vertical) shifts.

We find that mean Time Variance in vertical shift sequences is about 25% higher than that in horizontal shift sequences. This supports our hypothesis about inherent anisotropic sensitivity of DINOv2 features in case of horizontal or vertical shift motion. We will include this analysis on synthetic sequences along with qualitative tSNE visualizations in the supplementary material of the final paper.

It is worth asking why this difference is observed in horizontal vs vertical motion. Is it something to do with the DINO's training procedure (e.g., cropping mechanism) or position encodings in DINO or something else? Likewise, this connects to how we remedy this (e.g., by training DINO with rotated images?). All these questions require more time and deeper investigation and we defer them to future work. However, we do offer three directions that improve the performance on horizontal motion.

Possible mitigation. We highlight three promising directions to fix this. In the following, we show the resulting improvements in Table 11, and then explain the rationale for each direction below.

- Scaling up the image resolution at test-time: we hypothesize that encoding of fine-grained information such as the spatial positions of objects should improve with image resolution. This provides a +7.2 point improvement in the spatial position while improving the average across all types of changes by 2.2%.
- Improving the base model: an inherently better model should encode spatial positions better. We use WebSSL [2] which is a scaled up DINO-like image model trained on 2B samples. It yields a boost of +4.5% on horizontal shift.
- Using image rotations as a form of test-time recovery: interestingly, we note that encoding of position along the vertical axis is better than that along the horizontal axis. We exploit this fact and concatenate embeddings of videos rotated by $\pi/2, \pi, 2\pi/3$ with that of the upright video.

There is still a gap of 14% between best LiFT model and VideoMAE on horizontal shift. There is more work to do here but we re-iterate that LiFT is stronger than the video encoders for all other kinds of visual change.