arXiv:2401.10805v2 [cs.CV] 26 Apr 2024

Learning to Visually Connect Actions and their Effects

Eric Peh, Paritosh Parmar, Basura Fernando

Institute of High-Performance Computing, Agency for Science, Technology and Research, Singapore

Abstract. In this work, we introduce the novel concept of visually Connecting Actions and Their Effects (CATE) in video understanding. CATE can have applications in areas like task planning and learning from demonstration. We identify and explore two different aspects of the concept of CATE: Action Selection and Effect-Affinity Assessment, where video understanding models connect actions and effects at semantic and fine-grained levels, respectively. We observe that different formulations produce representations capturing intuitive action properties. We also design various baseline models for Action Selection and Effect-Affinity Assessment. Despite the intuitive nature of the task, we observe that models struggle, and humans outperform them by a large margin. The study aims to establish a foundation for future efforts, showcasing the flexibility and versatility of connecting actions and effects in video understanding, with the hope of inspiring advanced formulations and models.

1 Introduction

Actions can transform systems, scenes, and environments. Humans can connect their actions with the corresponding effects, *i.e.*, given an initial state[1], when some action is applied, we can imagine the outcome or the resultant state of the system—*e.g.*, consider a scenario where a person is holding an object; if they release their grip, we intuitively know that the object will inevitably fall to the ground. This understanding of causal mechanisms is crucial as it allows humans to plan and execute goal-directed actions. Humans can anticipate the consequences of their actions, make informed decisions, and adjust their behaviour. The integration of visual information with motor actions allows individuals to build a coherent and adaptive representation of the world, enabling them to navigate their environment effectively. The presence of such capabilities in video understanding systems and autonomous robots is essential, with applications spanning areas such as task planning and learning from demonstration. This not only makes them versatile but also enables real-time adjustments to their behaviour. Despite being recognized as a fundamental aspect of human cognition, the capability to link actions and their physical effects remains largely unexplored in the existing body of literature on video understanding. Towards that end, in this work, we introduce a novel concept of visually connecting actions and their effects.

Our work makes contributions to the understanding of the relationship between actions & effects by uncovering and investigating two key aspects: action selection &

[1] In general, the state of the system before the action is applied is termed as initial state, and the state of the system after the action is applied is termed the final state.

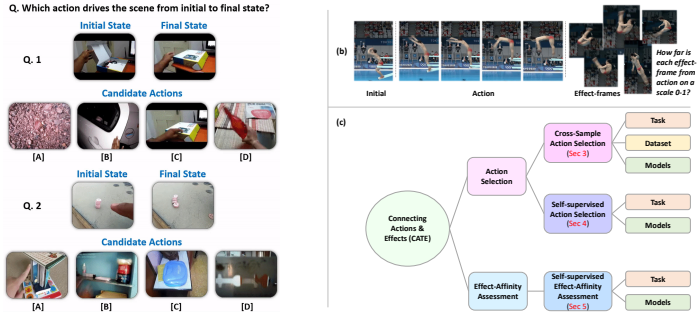


Fig. 1: Concept. (**Left**) *Please view in AdobeReader to play the embedded videos for better explanation.* We have shown two exemplary Action Selection-questions, along with candidate action-options. Correct answers are (B) (not (C)) & (A) in questions 1 & 2, resp. (**b**) Effect-Affinity Assessment aspect of connecting actions and effects (frames cropped only for concept demonstration purposes; in practice, whole frames used). (**c**) Paper organization guide.

Effect-Affinity Assessment. These aspects offer distinct perspectives on connecting actions to their effects in visual scenes. Connecting actions & their effects can be approached from various perspectives, leading to different task formulations. For instance, actions & effects could be linked at a semantic level. Following this perspective, we introduce the task of Action Selection, which requires video understanding models to visually connect actions & their effects. Video understanding models are presented with a pair of initial & final states of a scene, & it must select the correct action from a set of options that transitions the system from the initial state to the final state (ref. Figure 1).

Now, let us take a look at connecting actions and effects at a finer granularity. To achieve this, we introduce the task of Effect-Affinity Assessment, where a video understanding model is tasked with inferring how closely or directly related is an effect to an action (Figure 1, Figure 3). So, while Action Selection requires connecting actions and effects at a semantic level with coarser granularity, Effect-Affinity Assessment involves discerning at a finer resolution.

Interestingly, we observe that different formulations of connecting actions and their effects learn representations that capture intuitive properties of actions. For example, the Action Selection model learns to track without any explicit supervision (see Figure 4); while Effect-Affinity Assessment representations learn to encode the pose of humans without explicit supervision (see Figure 6b). We also observe that connecting actions and effects can serve as an effective self-supervised pretext task to learn generic video representations from unlabeled videos. Furthermore, we note that the ability to generate effect can benefit other tasks such as action anticipation.

While connecting actions and their effects may seem an easy task for humans, we found it to be quite challenging for state-of-the-art video understanding models. To this end, we propose several baseline frameworks that improve over the current state-of-the-art video understanding models, but we find that there is still a large performance gap between humans and machines on this task.

We aim to lay a solid foundation for future efforts by exploring different aspects of connecting actions and effects and devising task formulations and models to realize these aspects. Multiple formulations show that connecting actions and their effects is a flexible and versatile overarching concept with many possible applications. Developing and evaluating a diverse set of baseline models/solutions helps ensure comprehensive coverage of the problem space and provides a more robust evaluation framework. We hope that these will inspire future efforts to develop more advanced formulations and models with further applications and better performance.

The paper is organized as follows. We start by reviewing related work in Section 2, followed by task formulations in Sections 3, 4, 5. For better organization, Models and Datasets are discussed along with their corresponding tasks. Task and models are evaluated and validated in Section 6; finally, concluding the paper in Section 7.

2 Related Work

Video understanding tasks. Video understanding is a mature branch of computer vision addressing action analysis problems such action recognition [3, 11, 26, 37, 41], action segmentation/localization [5, 7, 20, 35], and action quality assessment [1, 31, 32, 49]. While tremendous progress has been made in these directions, connecting actions and their effects largely remains to be addressed. Towards that end, in this work, we study linking actions and their effects and study consequences of the same. We make use of the techniques and insights gleaned from existing video understanding tasks towards the problem of connecting actions and effects.

Viewing actions as transformations. Actions can be viewed as transformations or agents of transformations occurring in the scene or an environment [29, 38, 44]. Based on this, Wang *et al*. [44], proposed to model actions as fixed class-specific transformation matrices, which when applied to a precondition would produce the effect. Soucek *et al*. [38] proposed to learn action-specific models to localize states and actions. Similar to actions, Nagarajan and Grauman [29] view object properties as operators on objects. However, these works either focus on learning a rigid set of class-specific transformation matrices [44], or learning only action-specific models [38] or limited to static images [29, 38] and do not focus on studying action dynamics and their effects. Our work studies linking action dynamics and effects in much more depth and breadth. Our task also demonstrates wider applicability. We further take inspiration of such prior work and develop their dynamic counterparts which can generate transformations on the fly from actions and that way help in connecting actions and their effects.

Synthetic datasets. Obtaining the ground-truth for object properties and positions can be made easier and more efficient with the help of synthetically generated datasets [10, 19]. As such, synthetic datasets form a lucrative option for studying state-changes as done in [16]. However, sophisticated approaches developed towards such synthetic datasets are not generally applicable to real-world scenarios where the models do not have access to a high level of control over the attributes of states and actions. Our work conducts all the studies and validating experiments on real-world datasets. Furthermore, our novel cross-sample setting can be seen as a real-world equivalent to View

transformation [16] or even more challenging as many other factors are varied. Another difference between our work and those using synthetic data is that they do not contain actions involving humans, while ours has been experimented specifically with human actions in the real-world.

Modeling state-changes. A series of works [18, 21, 30] address the problem of computing state changes from the initial and final states of the environment. Computing state-changes is an important part of our proposed problem. However, unlike our work, these works do not link the changes in state with the actions driving those changes.

Assorted work on visual reasoning. Liang *et al*. [23] study an interesting problem of coming up with likely explanations for partial observations. Parallels could be drawn between initial and final states and partial observations. However, their work is concerned with forming semantically higher-level explanations, whereas ours is more focused on foundational-level connection—linking action mechanics or kinematics with their effects. Other interesting reasoning problems have been introduced in recent years [14, 45], but they are orthogonal to ours.

3 Task Formulation for Action Selection

Task formulations are crucial as each one has a unique objective and enables control over the model's focus to learn representations that capture slightly different types of visual features. We start by discussing the first task formulation—Action Selection.

Objective. Given the initial and desired final states, this task formulation requires video understanding models to select the correct action instance that changes the scene from its initial to its final state. Taking inspiration from visual question-answering literature, we cast the problem as a multiple-choice question answering; and provide four action-video choices for answers—see Figure 1. From a total of four choices/options, only one is correct. The task is to select the correct action-video. Action Selection task can enable Autonomous Agent/Robot Action Selection from human demonstrations from existing large-scale video datasets—discussed in subsection 3.1.

Constructing *Cross-Sample* question-answer sets. To form a question, we divide a video instance of a human performing an action into the initial state of the scene (starting few frames), & the resultant final state (last few frames). The remaining middle frames after keeping a certain margin from the initial & final state frames become the correct action option. However, the correct answer option can be drawn in two ways: 1) same-sample setting: from the same video clip as the states as explained earlier; or 2) cross-sample setting: from an altogether different instance, but the action category remains the same as the video from which states are drawn. Cross-sample setting can be viewed as a very strong data augmentation because essentially all factors such as background, colors, viewing angle, actors, & scene setup may be different from state-video, while semantic action class remains the same. This setup enables the isolation of action semantics from irrelevant features of the scene & demands to focus on the relevant parts of the video, thereby helping connect actions & their effects effectively.

3.1 Dataset

We introduce a novel dataset to support our proposed Action Selection task. Each sample in our dataset consists of the “question", and the corresponding a single correct answer and three incorrect answers.

Dataset sources. The first challenge in creating our dataset was choosing appropriate large-scale video datasets containing numerous action classes involving significant state changes. For this, we chose: 1) Something-Something version 2 (SSv2) dataset [11]; and 2) COIN dataset [40]. SSv2 encompasses a diverse range of actions—physical movements, object handling, transformations, and interactions. It contains actions related to the manipulation of physical objects, such as lifting, pushing, pouring, stacking, and more, as well as actions that involve altering the position and shape of objects. COIN dataset on the other hand primarily centers on the execution of specific tasks and procedures involved in assembling, installing, building, or manipulating various objects and materials. It often comprises step-by-step instructions for tasks like assembly, installation, cooking, and maintenance, with an emphasis on precise and ordered procedures, including actions like “installing," “cutting," “putting," “removing," and “filling". Overall, using SSv2 and COIN helps us build a very diverse dataset.

Selecting action classes. Not all action classes from SSv2 & COIN involve significant state changes, & consequently, might be less suitable for CATE Action Selection task. For example, action classes like “approaching something with camera" are less likely to involve intended state changes; & as such should be discarded. To mitigate this, we manually filter usable action classes from these base datasets. To determine if an action class is usable, we first take into consideration the name of the action class. We further inspect randomly sampled videos from the action classes to verify if the action class contains significant intended state changes. Based on these two evaluations, we decide if the action class is usable. Overall, a combined 204 action classes are found to be usable. No overlap between action classes was ensured to avoid confusing the models.

Temporal annotations for states and actions. To effectively capture states and actions, we utilize the first and last 10% of all video frames as the initial and final states, respectively. The middle 80% represents the driving action, allowing flexibility for varying video lengths using percentages rather than fixed frame counts. Given potential noise in crowd-sourced datasets like SSv2, where clips may include irrelevant or shaky footage, we address this by detecting objects and hands. We retain video footage only after hands or relevant objects are detected and stop retention when they are not detected. This approach enhances dataset quality, as illustrated in supplementary.

Constructing quadruplets. We follow the cross-sample setting as mentioned in section 3 to construct quadruplets of question, correct answer, & incorrect answer options.

Causal-Confusion Hard Counterfactuals. Incorrect action options can be created by simply choosing drawing them from altogether a different action category than the category of the correct action option. While this strategy satisfies the requirement, we hypothesize that harder-to-distinguish incorrect options can be generated. Towards this

end, we introduce the Causal-Confusion hard counterfactual/incorrect action option generation strategy. For this we take the correct action option and apply one of the following transformations: 1) temporal-reversal—temporally reverse the action clip; 2) static—repeat the first action frame, which causally means that no real action is carried out; 3) flipping the video horizontally—actions moving something left to right will have completely opposite meaning. Note that with this strategy, samples can be generated that look identical to the correct action class, but differ in terms of causal factors. As such, this strategy encourages the video understanding model to focus on causal factors, while learning to be invariant to irrelevant factors such as background and appearance.

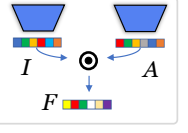
3.2 Baseline models

As the proposed task is new, and there are no established methods or benchmarks available for tackling this specific task, we design and explore a wide variety of baseline methods to connect actions and effects. For simplicity, only the best-performing methods are presented here, remaining methods are included in supplementary material.

Preliminaries.

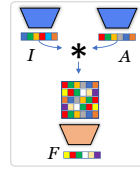
- *Action & State Representations.* Actions are represented as video clips, while states by very short clips. Unless specified otherwise, we use a pre-trained backbone model from a large-scale action recognition dataset to extract features for actions and states. Subsequently, various modules mentioned in the following are trained on these features.
- *Similarity vs Counterfactual reasoning.* Our methods involve feature matching in some form during training. Matching could be done: 1) just based on the similarity with the correct output; or 2) via counterfactual/contrastive reasoning by using similarity along with contrasting with incorrect outputs. In our experiments, we found counterfactual reasoning to work significantly better than similarity-based reasoning. Thus, unless mentioned otherwise, assume that counterfactual reasoning is employed

Naive matching. As a simple baseline, in naive matching the average of the initial and final state features are matched with the action features using cosine similarity. The action answer with the highest similarity score is selected as the correct answer. This model involves no training, pretrained action recognition model is used as the state and action feature extractor.

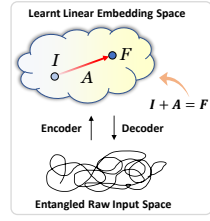


Treating actions as transformations. Actions can be viewed as transformations [44] which when applied to the initial state yields the final state. While the original approach uses fixed transformations [44], our adaptation facilitates conditional transformations computed on the fly. We develop a model in which a transformation vector is computed from an action clip (I) using a dense network. This transformation vector or matrix (A) when multiplied with the initial state vector yields the final state vector F. During training, we use contrastive learning [4], where the cosine similarities

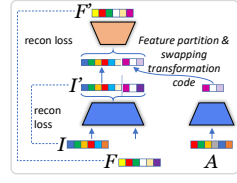
between the predicted and wrong action options are minimized while the similarity between predicted and correct action options is maximized. During inference, the action that produces the highest similarity with the final state feature is selected as the correct answer. We term these models as Actions as transformations (AT).



Modeling rich interactions between initial state & action (MoRISA). Here a bilinear model [24] is leveraged to capture rich interactions between the initial state and the action to generate the final state in feature space. Essentially, initial state (I) and action representations (A) are fed into a bilinear model, which outputs a final state representation (F). The parameters of the model are trained using contrastive learning as before and the inference is done similarly to AT model above.

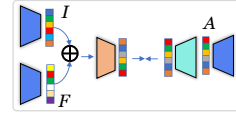


Learning linear embedding State-Action space. In this framework, actions and initial states undergo transformation into a linear embedding state-action representation space through a Transformer encoder. The hypothesis suggests that within this linear, disentangled space, *adding* an action vector to the initial state representation enables a transition to the final state in the representation space. The Transformer encoder processes a sequence of initial state and action representation vectors, and the resulting output is decoded by a linear decoder to produce the final state feature vector. During training, all model parameters are optimized end-to-end with the objective of aligning the resultant final state representation with the ground truth final state representation. Optimization and inference process remains the same as AT model. We call this model AEXformer.



Explicitly Disentangling State-Transformation. We will start by presenting method overview, followed by details. *Method overview:* all model (encoders and decoders) parameters are trained end-to-end by enforcing the following. 1) **Transformation Encoding.** The initial and final states are transformed into an initial state and a corresponding state-transformation code through an encoder. This transformation code captures the changes or effects between the initial and final states. 2) **Reconstruction using Decoder.** Given the initial state and the state-transformation code, a decoder reconstructs the final state. This decoder essentially learns to generate the final state based on the initial state and the transformation code. 3) **Obtaining Transformation Code from Cross-sample Actions.** Additionally, a state-transformation code can be obtained from the action (but from another sample) occurring between the initial and final states. This is done through another encoder that distills the transformation code from the action. Note that, during training the final state is reconstructed using both transformation codes—obtained from states and action. By obtaining equivalent state-transformation codes both from the states directly and from the actions, the model learns to connect actions to their effects. The decoder then utilizes this transformation code to reconstruct the final state from the initial state, effectively capturing the impact of actions on the state transition in the video data. *Method details:* To explicitly isolate a n-dimensional "transformation

"code", we pass the initial and final states (each represented by m-dimensional features) through an encoder, which outputs a $(m+n)$-dimensional vector. We enforce the condition that the first m elements of this vector represent the initial state and the remaining n elements represent the transformation "code". Now, using another encoder, we distill a n-dimensional transformation code from an action option. We swap the transformation code obtained using states with that obtained from the action option. A decoder then reconstructs the final state from the extracted initial state and swaps the transformation code. All encoders and decoders are trained by enforcing that the extracted initial state be similar to the ground truth initial state and the reconstructed final state be similar to the ground truth final state in case the transformation code was coming from the correct action option (minimize the similarity if the transformation code was from incorrect action option). This approach is inspired by [31]. However, while their approach disentangles human pose and appearance, our approach extends it to disentangle state-transformation for linking actions and their effects.



Analogical Reasoning. In this model, an analogy is made between state-changes or effects from one instance and the action from another instance. To accomplish that, the framework consists of a state change computer (SCC) head, and an action encoder head. SCC head takes in the scene information abstracted by backbone from initial and final states; and analyzes the two sets of features and computes the state-change occurring between the two states. To do this, the SCC head concatenates the two sets of features and processes them through a multi-layer perceptron, yielding a compact feature vector representing the state change. Operations like addition and subtraction can be alternatives to our deep state-change computer. However, we hypothesize that such operations are more likely to work where camera motion is absent or minimal. In the presence of camera motion, the natural pixel-to-pixel correspondence no longer holds, and such hard-coded operations would underperform, while more flexible solutions like our deep SCC would be better able to handle such noisy cases. The Action encoder extracts information from action clip. During training, we bring close together the state change and the correct action representations, while pushing apart state change and incorrect action representations. We found alternating between optimizing state and action heads to improve the performance. We hypothesize that this process can be viewed as the SCC learning to imagine action from states and framework making an analogy between imagined action and actual action options. Thus, we term this model as an analogical reasoning model.

4 Task Formulation for Self-Supervised Action Selection

The concept of connecting actions and their effects is closely related to the idea of the sensorimotor loop and the reciprocal relationship between vision and movement/action in human development [8, 9]. This reciprocal relationship involves learning via self-supervision. Similarly, we hypothesize that useful action representations can be learnt by deep video understanding models via learning to visually connect actions and their effects in a self-supervised way from unlabeled videos.

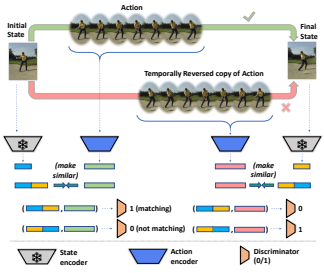


Fig. 2: Self-supervised pretext task based on connecting actions and effects. *Please zoom-in to view better.* Applying action can bring the scene from its initial state to the final state, while applying a temporally reversed version of it cannot. State-encoders are frozen; state-encoders and discriminator can be discarded after training—only the action encoders are retained for further usage.

Objective. Given an unlabeled video of some human action, the first and last frames are chosen to be representatives of the initial and the final states of a scene. The remaining middle portion of the video is considered as the action(-clip). Now, we design a self-supervised task based on the observation that applying action to the initial state would yield the final state; however, applying a temporally-reversed version of the same action to the initial state would not yield the given final state. In other words, the video understanding model is tasked with selecting the action which can take the scene from its initial state to final state—forward version of the action would, while temporally-reversed would not. We hypothesize in the process of solving this task, the model can learn useful video representations. This task has some resemblance with works like [6,28], however, these works randomly shuffle frames, while our task operates specifically on the {initial state, action, final state} triplet. With random shuffling, potentially low-level cues could be leveraged; while our formulation offers connecting and discerning actions and effects at a semantically higher level. The concept along with implementation framework is shown in Figure 2. Note that the framework is a slightly modified version of Analogical Reasoning model from § 3 and explained further below.

Framework. First, let us introduce two *ordered* paired representations: 1) forward state-change: {initial state representation, final state representation}—order-preserving concatenation of the state representations; 2) backward state-change: {final state representation, initial state representation}. To avoid feature collapse, state representations are extracted using a SSL-pretrained feature extractor [51]. Note that, using SSL-pretrained feature extractor for downstream SSL methods is an accepted common practice as done in, *e.g.*, [13,39,47]. During training the action network is tasked with making forward-action representation as similar to the forward state-change, and backwards-action representation similar to the backwards state-change. Once the action model is trained, it is used for downstream tasks such as action recognition. Discriminator is discarded after training.

5 Task Formulation for Self-Supervised Effect-Affinity Assessment

Objective. Effect-Affinity Assessment aims to achieve a more nuanced understanding by focusing on discerning subtle effects in close proximity and determining the "distance" of an effect-frame from the action applied to an initial state. This task is formulated based on the observation that, when a short action is applied to an initial

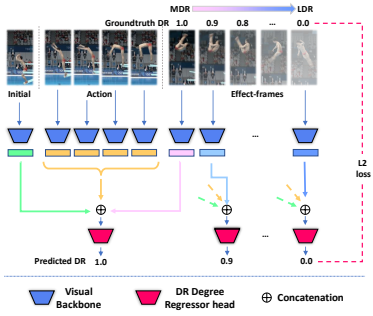


Fig. 3: Self-supervised Effect-Affinity Assessment. *Zoom-in to view better.* MDR: more directly related effect; LDR: less directly related effect. Degree of how directly related an effect is given on a scale of 1.0-0.0 written above effects. Farther the effect-frame from the action, lesser directly related it is to the action. Here we have shown cropped video frames to focus on the diver—in practice, we use the entire frame, and the network learns to focus on the diver through Effect-Affinity Assessment-based self-supervision.

state, the frame immediately following the completion of the short action represents the most direct effect. As the video progresses, subsequent frames gradually exhibit a diminishing connection to the effect of the initial short action. Note that, the distance of an effect frame being a measurable and observable aspect, is used as a proxy to reflect the strength of the connection that effect-frame to the action.

Framework. During training, the network learns to determine the "distance" (or equivalently, degree of relatedness, DR) of effect-frames from actions. Specifically, each training sample consists of an initial state, a short action clip, and a set of K temporally ordered frames following the action, which form effect-frames. Each of the K effect-frames are assigned a *proxy groundtruth* label which indicates the distance of an effect-frame from the end of the action clip. Specifically, in a series of K temporally-ordered frames, k-th frame is assigned a value of $1 - k/(K-1)$, where k ranges from 0 to $K-1$. This way, the closest or the most directly-related frame is assigned a proxy-label of 1, while the farthest effect-frame is assigned a proxy-label of 0. The task for the network is to predict these proxy-labels as accurately as possible. Framework is shown in Figure 3. Network parameters are optimized through a regression loss (L2) function. Note that ground-truth labels $(1 : 1/K : 0)$ are artificially generated and are arbitrary—any other scale can be used as long as it can express how far is an effect-frame from the action clip. We use unlabeled videos and artificially generated ground-truth to learn representations using this self-supervised pretraining task.

In the process of solving this task, we hypothesize that the model learns to focus on intricate details, such as identifying the specific maneuvers executed by the athlete and understanding the consequent sequence of poses. Additionally, it learns to discern how the athlete's position evolves as a direct outcome of these maneuvers. These details have an overlap with the elements of interest in the task of action quality assessment (AQA). Therefore, we hypothesize that representations learnt in solving our SSL CATE Effect-Affinity Assessment should help with downstream task of AQA. To validate this, the self-supervised model is tested on action quality score prediction task, where the model scores how well an action was performed. A brief introduction to AQA task and implementation details are provided in the supplementary materials.

6 Experiments

In this section, we conduct the following experiments:

- Benchmarking on Action Selection task
- Probing into where the models need to look or pay attention to solve Action Selection task
- Leveraging Action Selection task for improving on action anticipation task
- Self-supervised representation learning by learning to connect actions and effects
- Self-supervised Effect-Affinity Assessment for AQA
- Pose-encoding representations from solving self-supervised Effect-Affinity Assessment

6.1 Cross Sample Action Selection

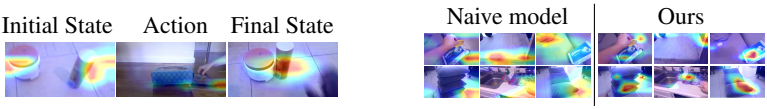
Implementation details. All the models are implemented using PyTorch [34], and optimized using ADAM optimizer [22] with a learning rate of 1e-4. We train all the models using CATE train set and select the best-performing iteration based on the validation set. The performance of the best model is then verified on the test set. Noting the state-of-the-art performance of Video Swin model [26] on various video understanding tasks, we use a Kinetics pretrained version of it as the backbone feature extractor. Note that CATE task and methods are not limited to video swin model, and future work may use other models.

Model	Perf.
Random	24.39
Naive	47.39
Actions as transformations	47.33
MoRISA	52.82
AEXformer	54.65
Explicit Disentangler	52.25
Analogical Reasoning	**55.20**
Humans	81.33

Table 1: Performance of various models at *Action Selection task.*

Answer evaluation. For the model's answer to be considered correct it has to, from the given options select only one option that matches the ground truth based on the highest similarity. In the scenario where all options return equal similarities, it is deemed to have failed to answer correctly. We use accuracy to evaluate the performance metric.

In this experiment, we measure the performance of various baseline approaches on the CATE task using the full CATE dataset. The results are summarized in Table 1. The Naive approach performs much better than random weights/chance accuracy. The approach treating actions as a transformation (AT) performs on par with the naive model. However, transformation-based approaches could be made more effective by simplifying the learning process and utilizing element-wise learnable weights for the interactions between the initial state and action vectors (MoRISA). We observe further improvements by learning a linear space for state-action interactions (AEXformer). Although it is a simple approach, it is found to be effective for connecting actions and effects. Learning to explicitly tease out or disentangle the 'transformation' from actions and states works better than naive and AT approaches; however, it performs slightly inferior to AEXformer. While both approaches can achieve disentanglement in some form, empirically we observe that implicitly disentangling in the process of learning a



(a) (b)

Fig. 4: Probing where the model pays attention when connecting action and its effects.. (a) Action is '*putting something close to something*'. (b) Where our model attends vs where action recognition model attends. Our model focuses on state changes and driving action and effects; notice the avocado and coin being tracked. Action recognition without reasoning module seems to be doing simple matching based on texture. *Please zoom in to view better*.

linear space might be a better alternative than explicit disentanglement via feature partition and swapping. The Analogical Reasoning model performed the best. However, it is worth noting that training it is a bit more complicated than other models, as we had to resort to alternating between optimizing state change encoders and action encoders. Nonetheless, we observed that it achieves peak performance faster than other models. AEXformer and Explicit Disentangler take the longest to achieve peak performance.

Human baseline. To put the performance of our baseline models into perspective, we conduct a study to assess human performance. We prepare a smaller subset of 100 questions and observe the performance of three undergraduate students on it. The average human performance is then compared with our model's performance on this subset—see Table 1. We observe that humans performed significantly better than all of our baseline models on this seemingly easy task of connecting actions and effects.

Qualitative results. We have visualized where our models "look" or focus when solving our visual action-effect task in Figure 4.

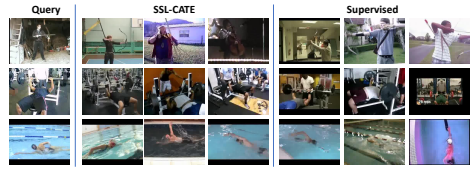
Action anticipation. We explore the utility of effect generation in action anticipation. Following the literature [50], we conduct an action anticipation experiment on the COIN dataset. The task is to predict the next action one second into the future. Drawing inspiration from [15], we hypothesize that being able to generate unseen futures can aid by providing missing information. Thus, we generate this missing information using our AEXformer, which is already trained on our CATE Action Selection dataset and then frozen. At this stage, we simply use it as a "future feature extractor." We then integrate this future feature with the current action feature and learn a simple decoder to predict the next action one second into the future. We also consider a case where we do not leverage the future feature. Frameworks for both approaches are shown in Supplementary. The top-5 mean recall results are presented in Table 2. It is evident that the future feature does help with the action anticipation task as well.

Model	Performance
Zatsarynna *et al.* [50]	13.39
Zatsarynna *et al.* [50] w Goal Consistency	13.93
Ours	18.80
Ours w generated effect	**19.02**

Table 2: Utility of generating effect in action anticipation task.

Model	Top-1	Top-5	Top-10	Top-20	Top-50
VCOP [48]	12.5	29.0	39.0	50.6	66.9
VCP [27]	17.3	31.5	42.0	52.6	67.7
MemDPC [12]	20.2	40.4	52.4	64.7	-
VideoPace [43]	20.0	37.4	46.9	58.5	73.1
SpeedNet [2]	13.0	28.1	37.5	49.5	65.0
PRP [49]	22.8	38.5	46.7	55.2	69.1
Temp.Tran. [17]	26.1	48.5	59.1	69.6	82.8
TCGL [25]	22.4	41.3	51.0	61.4	75.0
STS [42]	39.1	**59.2**	**68.8**	**77.6**	**86.4**
Ours SSL-CATE	**41.5**	56.2	64.7	73.5	83.7

(a)



(b)

Fig. 5: (a) Nearest Neighbor retrieval quantitative results on UCF101. Best performing method is **bold**-faced, and the second best method is underlined. **(b) Nearest Neighbor retrieval qualitative results.** Visualized Top-3 retrievals on UCF101.

6.2 Learning visual representations from unlabeled videos using CATE

After pretraining with the self-supervised action selection task (section 4) on the UCF101 [37] train set without using any labels, we freeze the backbone (C3D architecture [41]). No fine-tuning is performed. The frozen backbone is then used as a feature extractor. Following standard practice in the literature [17, 43], we conduct a nearest neighbor retrieval experiment. Quantitative results for the nearest neighbor retrieval experiment are presented in Figure 5a. Our CATE model is compared with other state-of-the-art SSL video representation learning approaches. We observe that SSL-CATE performs the best overall in terms of Top-1 accuracy, ranking second best in the remaining metrics, only outperformed by STS. These results highlight the significance of CATE in effectively supporting the learning of general action representations.

We further probe into what the model might be learning by retrieving videos similar to the randomly chosen query videos. Results are presented in Figure 5b. We observe that the retrieved results are mostly from the correct action class, suggesting that CATE encourages the model to learn the tracking of important human body parts and objects. Upon further investigation of some cases where the retrieval is from the incorrect action class, we found that motion patterns in the retrieved videos can potentially result in the same state change as the action in the query videos. For example, in archery and playing cello, pulling the arrow in archery and moving the bow/stick on the cello involve similar hand movements from humans. We further compare the results with a fully-supervised model. Qualitatively, we found that the CATE-pretrained retrieved results are on par with a model trained with full action class label supervision.

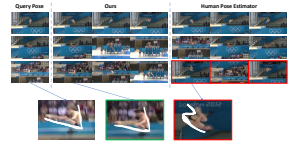
6.3 SSL Effect-Affinity Assessment for AQA

In this case, we train using self-supervision grounded in connecting actions and their effects, as presented in section 5. Following the literature [31, 36], we carry out SSL training on the training set of the MTL-AQA dataset [33]. After SSL training, we do not perform any fine-tuning; we simply use the backbone as a feature extractor. Features are extracted from 16 uniformly sampled frames, and then a linear regressor is applied on top of them. We compare SSL CATE-Effect-Affinity Assessment with other self-supervised models, and the results are shown in Figure 6a. We observe that SSL-CATE

Effect-Affinity Assessment self-supervision outperforms other state-of-the-art methods, indicating that CATE-Effect-Affinity Assessment does help in learning features suitable for fine-grained action analysis tasks such as action quality assessment.

Model	Performance
SSL Action Alignment [36]	77.00
SSL Motion Disentangler [31]	77.63
Ours SSL CATE Effect-Affinity Assessment	**79.36**

(a) Performance evaluation.



(b) Pose retrieval results.

Fig. 6: (a) Performance evaluation on action quality assessment task on MTL-AQA dataset. (b) Pose retrieval results. Top-3 retrievals containing diver using our model vs [46]. In this experiment, what we are looking for is if the retrieved results have the athlete in similar pose and orientation as the query pose (left most column). For explanation, we have drawn red bounding box around wrong retrievals. As highlighted with white-colored annotations in zoomed in view, in the wrong retrieval the athlete is in somersault tuck pose, while the query has the athlete in pike position. In the correct retrieval as done by our SSL-CATE-Effect-Affinity Assessment, the retrieval results have the athlete in the pike position.

SSL-CATE Effect-Affinity Assessment trained representations learn to encode human pose. In this experiment, we further probe into what the SSL-CATE Effect-Affinity Assessment network might be paying attention to. Using the trained model from the above experiment, features are extracted from diving video clip frames. Then, we carried out a pose retrieval experiment—given a frame containing a diver in a certain pose (query pose), the objective is to retrieve divers in a similar pose as the query pose. We further compare retrieval done using SSL-CATE trained representations against an off-the-shelf dedicated pose estimation model [46]. We find that CATE representations are better able to retrieve poses than a dedicated pose estimation model (Figure 6b). We hypothesize that this might be because the off-the-shelf pose estimator might not be able to handle motion blurring and divers in convoluted poses, while CATE-based self-supervision allows learning pose-sensitive robust representations directly from challenging videos.

7 Conclusion

In this work, we address the underexplored area of visually connecting actions and their effects in video understanding. By introducing tasks like action selection and Effect-Affinity Assessment, we shed light on the challenges and opportunities in understanding causal mechanisms for machines. Despite efforts to enhance state-of-the-art models, a significant performance gap between human and machine capabilities persists in this task. The study lays a solid foundation for future research, emphasizing the flexibility and versatility of connecting actions and effects. We hope that these insights will inspire the development of advanced formulations and models, paving the way for improved performance and broader applications in the field of video understanding.

Acknowledgements. This research/project is supported by the National Research Foundation, Singapore, under its NRF Fellowship (Award# NRF-NRFF14-2022-0001).

References

1. Bai, Y., Zhou, D., Zhang, S., Wang, J., Ding, E., Guan, Y., Long, Y., Wang, J.: Action quality assessment with temporal parsing transformer. In: European Conference on Computer Vision. pp. 422–438. Springer (2022) 3
2. Benaim, S., Ephrat, A., Lang, O., Mosseri, I., Freeman, W.T., Rubinstein, M., Irani, M., Dekel, T.: Speednet: Learning the speediness in videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9922–9931 (2020) 13
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017) 3
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020) 6
5. Ding, G., Sener, F., Yao, A.: Temporal action segmentation: An analysis of modern techniques. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023) 3
6. Fernando, B., Bilen, H., Gavves, E., Gould, S.: Self-supervised video representation learning with odd-one-out networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3636–3645 (2017) 9
7. Ghoddoosian, R., Sayed, S., Athitsos, V.: Hierarchical modeling for task recognition and action segmentation in weakly-labeled instructional videos. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1922–1932 (2022) 3
8. Gibson, E.J., Pick, A.D.: An ecological approach to perceptual learning and development (2000) 8
9. Gibson, J.J.: The ecological approach to visual perception: Classic edition (2014) 8
10. Girdhar, R., Ramanan, D.: Cater: A diagnostic dataset for compositional actions and temporal reasoning. arXiv preprint arXiv:1910.04744 (2019) 3
11. Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al.: The" something something" video database for learning and evaluating visual common sense. In: Proceedings of the IEEE international conference on computer vision. pp. 5842–5850 (2017) 3, 5
12. Han, T., Xie, W., Zisserman, A.: Memory-augmented dense predictive coding for video representation learning. In: European conference on computer vision. pp. 312–329. Springer (2020) 13
13. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020) 9
14. Hessel, J., Hwang, J.D., Park, J.S., Zellers, R., Bhagavatula, C., Rohrbach, A., Saenko, K., Choi, Y.: The abduction of sherlock holmes: A dataset for visual abductive reasoning. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVI. pp. 558–575. Springer (2022) 4
15. Hoffman, J., Gupta, S., Darrell, T.: Learning with side information through modality hallucination. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 826–834 (2016) 12
16. Hong, X., Lan, Y., Pang, L., Guo, J., Cheng, X.: Transformation driven visual reasoning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6903–6912 (2021) 3, 4

17. Jenni, S., Meishvili, G., Favaro, P.: Video representation learning by recognizing temporal transformations. In: European Conference on Computer Vision. pp. 425–442. Springer (2020) 13
18. Jhamtani, H., Berg-Kirkpatrick, T.: Learning to describe differences between pairs of similar images. arXiv preprint arXiv:1808.10584 (2018) 4
19. Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., Girshick, R.: Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2901–2910 (2017) 3
20. Kalogeiton, V., Weinzaepfel, P., Ferrari, V., Schmid, C.: Action tubelet detector for spatio-temporal action localization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4405–4413 (2017) 3
21. Kim, H., Kim, J., Lee, H., Park, H., Kim, G.: Agnostic change captioning with cycle consistency. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2095–2104 (2021) 4
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 11
23. Liang, C., Wang, W., Zhou, T., Yang, Y.: Visual abductive reasoning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15565–15575 (June 2022) 4
24. Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear cnn models for fine-grained visual recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 1449–1457 (2015) 7
25. Liu, Y., Wang, K., Liu, L., Lan, H., Lin, L.: Tcgl: Temporal contrastive graph for self-supervised video representation learning. IEEE Transactions on Image Processing **31**, 1978–1993 (2022). https://doi.org/10.1109/TIP.2022.3147032 13
26. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video swin transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3202–3211 (2022) 3, 11
27. Luo, D., Liu, C., Zhou, Y., Yang, D., Ma, C., Ye, Q., Wang, W.: Video cloze procedure for self-supervised spatio-temporal learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 11701–11708 (2020) 13
28. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and learn: unsupervised learning using temporal order verification. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. pp. 527–544. Springer (2016) 9
29. Nagarajan, T., Grauman, K.: Attributes as operators: factorizing unseen attribute-object compositions. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 169–185 (2018) 3
30. Park, D.H., Darrell, T., Rohrbach, A.: Robust change captioning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4624–4633 (2019) 4
31. Parmar, P., Gharat, A., Rhodin, H.: Domain knowledge-informed self-supervised representations for workout form assessment. In: European Conference on Computer Vision. pp. 105–123. Springer (2022) 3, 8, 13, 14
32. Parmar, P., Morris, B.: Action quality assessment across multiple actions. In: 2019 IEEE winter conference on applications of computer vision (WACV). pp. 1468–1476. IEEE (2019) 3
33. Parmar, P., Morris, B.T.: What and how well you performed? a multitask learning approach to action quality assessment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 304–313 (2019) 13

34. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf 11
35. Rahaman, R., Singhania, D., Thiery, A., Yao, A.: A generalized and robust framework for timestamp supervision in temporal action segmentation. In: European Conference on Computer Vision. pp. 279–296. Springer (2022) 3
36. Roditakis, K., Makris, A., Argyros, A.: Towards improved and interpretable action quality assessment with self-supervised alignment. In: The 14th PErvasive Technologies Related to Assistive Environments Conference. pp. 507–513 (2021) 13, 14
37. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012) 3, 13
38. Souček, T., Alayrac, J.B., Miech, A., Laptev, I., Sivic, J.: Look for the change: Learning object states and state-modifying actions from untrimmed web videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13956–13966 (2022) 3
39. Sun, C., Baradel, F., Murphy, K., Schmid, C.: Learning video representations using contrastive bidirectional transformer. arXiv preprint arXiv:1906.05743 (2019) 9
40. Tang, Y., Ding, D., Rao, Y., Zheng, Y., Zhang, D., Zhao, L., Lu, J., Zhou, J.: Coin: A large-scale dataset for comprehensive instructional video analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1207–1216 (2019) 5
41. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 4489–4497 (2015) 3, 13
42. Wang, J., Jiao, J., Bao, L., He, S., Liu, W., Liu, Y.H.: Self-supervised video representation learning by uncovering spatio-temporal statistics. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021) 13
43. Wang, J., Jiao, J., Liu, Y.H.: Self-supervised video representation learning by pace prediction. In: ECCV (2020) 13
44. Wang, X., Farhadi, A., Gupta, A.: Actions˜ transformations. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 2658–2667 (2016) 3, 6
45. Wu, B., Yu, S., Chen, Z., Tenenbaum, J.B., Gan, C.: Star: A benchmark for situated reasoning in real-world videos. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2) (2021) 4
46. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: European Conference on Computer Vision (ECCV) (2018) 14
47. Xiao, F., Liu, H., Lee, Y.J.: Identity from here, pose from there: Self-supervised disentanglement and generation of objects using unlabeled videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019) 9
48. Xu, D., Xiao, J., Zhao, Z., Shao, J., Xie, D., Zhuang, Y.: Self-supervised spatiotemporal learning via video clip order prediction. In: Computer Vision and Pattern Recognition (CVPR) (2019) 13
49. Yao, Y., Liu, C., Luo, D., Zhou, Y., Ye, Q.: Video playback rate perception for self-supervised spatio-temporal representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6548–6557 (2020) 3, 13
50. Zatsarynna, O., Gall, J.: Action anticipation with goal consistency. In: 2023 IEEE International Conference on Image Processing (ICIP). pp. 1630–1634. IEEE (2023) 12

51. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. arXiv preprint arXiv:2103.03230 (2021) 9