A Deeper Dive Into What Deep Spatiotemporal Networks Encode: Quantifying Static vs. Dynamic Information

Matthew Kowal[1,2], Mennatullah Siam[1], Md Amirul Islam[2,3]
Neil D. B. Bruce[2,5], Richard P. Wildes[1,4], Konstantinos G. Derpanis[1,2,4]
[1]York University, [2]Vector Institute for AI, [3]Ryerson University, [4]Samsung AI Centre Toronto, [5]University of Guelph
{m2kowal,msiam,wildes,kosta}@eecs.yorku.ca, mdamirul@ryerson.ca, brucen@uoguelph.ca

Abstract

*Deep spatiotemporal models are used in a variety of computer vision tasks, such as action recognition and video object segmentation. Currently, there is a limited understanding of what information is captured by these models in their intermediate representations. For example, while it has been observed that action recognition algorithms are heavily influenced by visual appearance in single static frames, there is no quantitative methodology for evaluating such static bias in the latent representation compared to bias toward dynamic information (*e.g. *motion). We tackle this challenge by proposing a novel approach for quantifying the static and dynamic biases of any spatiotemporal model. To show the efficacy of our approach, we analyse two widely studied tasks, action recognition and video object segmentation. Our key findings are threefold: (i) Most examined spatiotemporal models are biased toward static information; although, certain two-stream architectures with cross-connections show a better balance between the static and dynamic information captured. (ii) Some datasets that are commonly assumed to be biased toward dynamics are actually biased toward static information. (iii) Individual units (channels) in an architecture can be biased toward static, dynamic or a combination of the two.* [1]

1. Introduction

This paper focuses on the problem of interpreting the information learned by deep neural networks (DNNs) trained for video understanding tasks. Interpreting deep spatiotemporal models is a largely understudied topic in computer vision despite their achieving state-of-the-art performance on video understanding tasks, such as action recognition [53] and video object segmentation [48]. These models are trained in an end-to-end fashion to learn discriminative static and dynamic features over space and time. Here, we use the term *static* to refer to attributes that can be extracted from a single image (*e.g.* color and texture) and the term *dynamic* to attributes that arise from consideration of multiple frames (*e.g.* motion and dynamic texture).

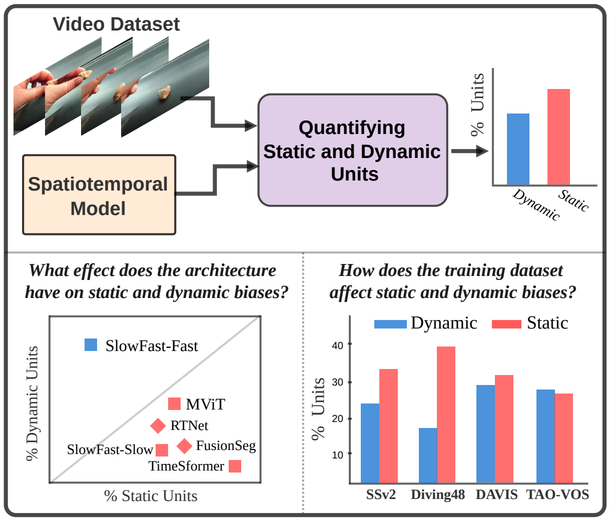

Figure 1. We introduce a general technique that, given a model and a video dataset, can quantify the bias in any intermediate representation within the model toward encoding static (red) or dynamic (blue) information. We use this technique to study the tasks of action recognition (squares) and video object segmentation (diamonds) and explore the effect of architectures and training datasets on static and dynamic biases.

While this learning-based paradigm has led to great success across a wide range of tasks, the internal representations of the learned models remain largely opaque. This lack of explainability is unsatisfying from both scientific and application perspectives. From a scientific perspective, there is limited understanding of what information is driving the decision-making underlying the network output. Elucidating the decision-making process may yield directions to improve models. From an applications perspective, there have been multiple cases showing the ethical and damaging consequences of deploying opaque vision models, *e.g.* [3, 21]. Currently, however, the explainability of

[1] Project page and code

spatiotemporal models is under-explored [25]. Some evidence suggests that these models exhibit considerable bias toward static information, *e.g*. [6,24,47]; therefore, an interesting question to answer about the representations in deep spatiotemporal models is: *How much static and dynamic information is being captured*? While a few video interpretation methods exist, they have various limitations, *e.g*. being primarily qualitative [16], using a certain dataset that prevents evaluating the effect of the training dataset [20] or using classification accuracy as a metric without quantifying a model's *internal* representations [20, 39].

In response, we present a quantitative paradigm for evaluating the extent that spatiotemporal models are biased toward static or dynamic information in their internal representations. We define bias toward a certain factor (dynamic or static) as the percentage of units (*i.e*. channels) within intermediate layers that encode that factor; see Fig. 1 (top). Inspired by previous work [10, 27], we propose a metric to estimate the amount of static vs. dynamic bias based on the mutual information between sampled video pairs corresponding to these factors. We explore two common tasks to show the efficacy of our approach as a general tool for understanding spatiotemporal models, action recognition and video object segmentation. We focus our study on answering the following three questions: (i) What effect does the model architecture have on static and dynamic biases? (ii) How does the training dataset affect these biases? (iii) What role do units that jointly encode static and dynamic information play in relation to the architecture and dataset?

Contributions. Overall, we make three main contributions. (i) We introduce a general method for quantifying the static and dynamic bias contained in spatiotemporal models, including a novel sampling procedure to produce static and dynamic video pairs. (ii) We propose a technique for identifying units that jointly encode static and dynamic factors. (iii) Using the aforementioned techniques, we provide a unified study on two widely researched tasks, action recognition and video object segmentation, with a focus on the effect of architecture and training dataset on a model's static and dynamic biases; see Fig. 1 (bottom). Among other findings, we discover in both tasks that all networks are heavily static biased, except for two-stream architectures with cross connections encouraging models to capture dynamics. Additionally, we confirm that, contrary to previous beliefs [2, 33], the Diving48 [33] dataset is not dynamically biased and Something-Something-v2 (SSv2) [19] is better suited to evaluate a model's ability to capture dynamics.

2. Related work

Interpretability of spatiotemporal models. Limited work has been dedicated to the interpretability of spatiotemporal models. Several efforts predicate model interpretation on proxy tasks, *e.g*. dynamic texture recognition [20] or future frame selection [18]. These approaches do not interpret the learned representations in the intermediate layers and in some cases require training to be performed on specific datasets [20]. Other work focused on understanding latent representations in spatiotemporal models either mostly concerned qualitative visualization [16] or a specific architecture type [51]. A related task is understanding the scene representation bias of action recognition datasets [33, 34]. However, these efforts did not focus on the effect of different architectural inductive biases on the learned intermediate representations. Our proposed interpretability technique is the first to *quantify* static and dynamic biases on *intermediate* representations learned in off-the-shelf models for multiple video-based tasks. Most prior efforts focused on a single task, and studied either datasets [33] or architectures [16, 35]. In contrast, our unified study covers six datasets and dozens of architectures on two different tasks, *i.e*. action recognition and video object segmentation.

Spatiotemporal models. Deep spatiotemporal models that learn discriminative features across space and time have proven effective for video understanding tasks [1, 48, 53]. Extant models can be broadly categorized (agnostic of the downstream task) into: two-stream approaches that separately model motion and appearance features [4, 14, 28, 38, 52], 3D convolutions that jointly model motion and appearance [4], attention-based models with different forms of spatiotemporal data association [2, 38], models relying on recurrent neural networks [43] and hybrid models that combine elements of the aforementioned models [4,38,43]. Our approach to quantifying bias is not limited to the particulars of a model and is applicable to all extant and future models. We empirically demonstrate the flexibility of our approach by evaluating a diverse set of models.

Action recognition. 3D convolutional networks are popular for learning spatiotemporal representations of videos for action recognition, *e.g*. [4, 22, 29, 41, 44]. Other work has considered two-stream architectures, where the dynamics were provided directly to one of the streams as optical flow, *e.g*. [15, 40]. Representative of the state of the art with convolutional networks is SlowFast [14], which is a two-stream 3D CNN that only takes RGB videos as input. To encourage each stream to specialize in capturing predominately static or dynamic information, the temporal sampling rates of the inputs to each stream differ. Recently, attention based approaches have proven to be suited to both static and time-series visual data, including action recognition, with variants of the transformer architecture [2, 12, 36, 45].

Video object segmentation. Deep video object segmentation (VOS) approaches can be categorized as automatic, semi-automatic and interactive [48]. In this work, we focus on automatic approaches that segment salient objects in videos, and the related task of motion segmentation [7]. We consider two-stream models that fuse motion and ap-

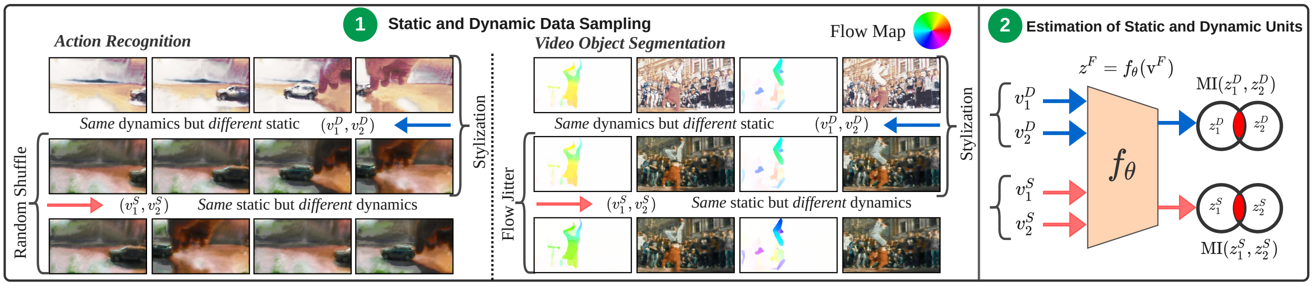
1
Static and Dynamic Data Sampling
Action Recognition
Same dynamics but different static
(v_1^D, v_2^D)
Stylization
Random Shuffle
(v_1^S, v_2^S)
Same static but different dynamics
Video Object Segmentation
Flow Map
Same dynamics but different static
(v_1^D, v_2^D)
Stylization
Flow Jitter
(v_1^S, v_2^S)
Same static but different dynamics
2
Estimation of Static and Dynamic Units
$z^F = f_\theta(\mathrm{v}^F)$
v_1^D
v_2^D
v_1^S
v_2^S
f_θ
$\mathrm{MI}(z_1^D, z_2^D)$
z_1^D
z_2^D
z_1^S
z_2^S
$\mathrm{MI}(z_1^S, z_2^S)$

Figure 2. Overview of our method for analysing bias towards static or dynamic information. We measure the dynamic and static biases in deep spatiotemporal models for two tasks: action recognition and video object segmentation. **(1)** We sample video pairs that share either *static*, (v_1^S, v_2^S), or *dynamic*, (v_1^D, v_2^D), information using video stylization [42] and frame shuffling or optical flow jitter (flow visualized in RGB format). **(2)** Given a pretrained model, f_θ, we compute the mutual information (MI) between intermediate representations of video pairs, z^F, to assess the model's bias toward either factor on a per-layer, l, or per-channel (*i.e.* unit) basis. In the supplement, we provide stylization examples in video format as well as additional static and dynamic samples.

pearance features. We also investigate the effect of no cross connections [28] relative to both motion-to-appearance [52] or bidirectional [38] cross connections.

3. Methodology

We introduce a novel approach to quantify the number of units (*i.e.* channels in a given layer) encoding static and dynamic information in spatiotemporal models; for an overview, see Fig. 2. Our approach consists of two main steps. First, given a number of pretrained spatiotemporal models on various datasets, we sample static and dynamic pairs of videos (Sec. 3.1). Second, we use these static and dynamic pairs to estimate the number of units in the model encoding each factor based on the mutual information shared between the pairs (Sec. 3.2).

3.1. Sampling static and dynamic pairs

Why static and dynamic? We define static as 'information arising from single frames' and dynamic as 'information arising from the consideration of multiple frames'. The main alternative attribute to dynamics that we considered was 'image motion' (*i.e.* trackable points or regions), but 'motion' is a subset of dynamic information [9, 50] (*e.g.* stationary flashing lights have dynamics but no motion). Thus, we consider dynamics over motion because it encompasses a wider range of visual phenomena. In complement, we choose the term 'static' over the possible alternative 'appearance', because dynamics also can provide appearance information, *e.g.* the contour of an object, even if camouflaged in a single frame, can be revealed through its motion. For our estimation technique, we produce video pairs that contain the same static information and perturbed dynamics, or vice versa, with the end goal of analyzing models trained on large-scale real-world datasets. We now detail our static and dynamic sampling techniques for both action recognition and VOS, as visualized in Fig. 2 (panel 1).

Action recognition. The action recognition models we consider take in multiple frames (four to 32). To construct video pairs with the *same* dynamics but *different* static information (*i.e. dynamic pairs*), we consider the same video but with two *different* video styles. For video stylization, we use a recent video stylization method (with four possible styles) that perturbs static attributes like color, pixel intensity and texture [42], but has less temporal artifacts (*e.g.* flicker) than stylization methods that consider each image independently [26]. These video pairs will contain objects and scenes that have identical dynamics, but have perturbed static information. To construct pairs with the *same* static information but *different* dynamics (*i.e. static pairs*), we take two videos of the same style, but randomly *shuffle* the frames along the temporal axis; see Fig. 2 (panel 1, left). In this case, the temporal correlations are altered while the static (*i.e.* per-frame) information remains identical.

Video object segmentation. The VOS models considered [28, 38, 52] take a single RGB frame and an optical flow frame as input to the appearance and motion streams, resp.; see Fig. 2 (panel 1, right). Therefore, we apply an alternative method to frame shuffling to obtain the *static* pairs. For the *static* pair, we use RGB images with the *same* style but alter the dynamics by jittering the optical flow. The RGB flow representation is used with hue and saturation encoding direction and magnitude, resp., and it is those parameters that we jitter. For the *dynamic* pairs, we use the *same* optical flow but a *different* image style. For creating stylized images, we use the same video stylization method noted above for action recognition [42], and then sample frames from the generated video.

3.2. Estimating static and dynamic units

We seek to quantify the number of units (*i.e. channels*) in a layer encoding *static* or *dynamic* information as well as the extent to which individual units perform static, dynamic or joint encodings. Inspired by recent work that focused on single images [10, 27], we use a mutual information estimator to measure the information shared between video pairs.

Layer-wise metric. Given a pre-trained network, f_θ, and a pair of videos, v_1^F and v_2^F, that share the semantic factor F (*i.e. static* or *dynamic*), we compute the features for an intermediate layer l as $z_1^F = f_\theta^l(v_1^F)$ and $z_2^F = f_\theta^l(v_2^F)$ (omitting the l on z to simplify the notation). We use $z_1^F(i), z_2^F(i)$ to denote the i^{th} unit (*i.e.* channel) in N^l dimensional features after a global average pooling layer. Our guiding intuition for this measurement is that units biased toward the *static* factor will result in a higher correlation among *static* pairs than the *dynamic* pairs and vice versa. Under the assumption that units in the intermediate representation $z_1^F(i), z_2^F(i)$ across the dataset are jointly Gaussian, the correlation coefficient can be used as a lower bound on mutual information [17, 30], as used in previous work [10, 27]. The number of units encoding factor F, N_F, is obtained by computing the correlation coefficient, S_F, over all N^l channels between all video pairs z_1^F, z_2^F, as

$$N_F = \sigma(\mathbf{S}) \cdot N^l = \frac{\exp(S_F)}{\sum\limits_{k=0}^{K} \exp(S_k)} \cdot N^l,$$

$$S_F = \sum_{i=1}^{N^l} \frac{\text{Covariance}(z_1^F(i), z_2^F(i))}{\sqrt{\text{Variance}(z_1^F(i))\ \text{Variance}(z_2^F(i))}}, \tag{1}$$

where we multiply the Softmax, $\sigma(\cdot)$, by the number of units in that layer, N^l, to compute the number of units encoding the semantic factor F relative to the other factors considered and $K = \{\text{static}, \text{dynamic}, \text{identical}\}$. In addition to *static* and *dynamic*, we consider a third factor in (1), the *identical* factor, where the video pairs have the same static and dynamic factors (*i.e.* same video, style, frame ordering and optical flow). This baseline factor is the correlation between the model's encoding of the same videos, that gives $S_{\text{Identical}} = 1$ for all layers.

Unit-wise metric. The correlation coefficient, S_F, estimates the relative amount of static and dynamic information over all units in a particular layer; note the pooling done by the summation *before* the Softmax in the layer-wise metric, (1). However, it is also desirable to measure static and dynamic information contained in each individual channel. This measurement allows for a more fine-grained analysis of how many channels (*i.e.* units) encode a factor F above a certain threshold, as well as identify any joint or residual (*i.e.* non-dynamic or static) units. Thus, we categorize each unit based on how much information (*i.e.* static vs. dynamic) is encoded, whether any units jointly encode both factors or if there are units that do not correlate with either type of information. We measure the amount of static and dynamic information encoded in each unit $i \in 1, \dots, N^l$ as

$$s_F^i = \frac{\text{Covariance}(z_1^F(i), z_2^F(i))}{\sqrt{\text{Variance}(z_1^F(i))\text{Variance}(z_2^F(i))}}, \tag{2}$$

where each s_F^i is the information of semantic factor F in unit i. Given these individual correlations, we calculate the individual factors by excluding the use of a Softmax and simply threshold the correlation for each factor with a constant parameter, λ, to yield our unit-wise metrics as

$$N_{\text{Joint}} = \sum_{i=1}^{N^l} \mathbb{1}[s_F^i > \lambda \forall F \in K]$$

$$N_{\text{F}} = \sum_{i=1}^{N^l} \mathbb{1}[s_F^i > \lambda \wedge s_k^i < \lambda \forall k \in K, k \neq F] \tag{3}$$

$$N_{\text{Residual}} = \sum_{i=1}^{N^l} \mathbb{1}[s_F^i < \lambda \forall F \in K],$$

where $K = \{\text{static}, \text{dynamic}\}$, N_{Joint} indicates units jointly encoding both and N_{Residual} are units not correlating with these factors under a certain threshold, λ. Note that we assign units to either joint, dynamic, static or residual and do not allow for an overlap to occur. This approach allows us to investigate the existence of units that jointly encode static and dynamic factors. For all experiments, we set $\lambda = 0.5$ since it is halfway between *no* and *full* positive correlation. The supplement has results with varying λ.

4. Experimental results

We choose the two tasks of action recognition and video object segmentation to demonstrate the generality of our approach. More specifically, they differ in their semantics (*i.e.* multi-class vs. binary classification), labelling (*i.e.* video-level vs. pixel-level), and input types (multi-frame images vs. single frame optical flow). We explore three main research questions and show the corresponding results with respect to our quantitative techniques for both tasks: (i) What is the effect of the model architecture on the *static* and *dynamic* biases (Sec. 4.1)? (ii) What effect does the training dataset have on *static* and *dynamic* biases (Sec. 4.2)? (iii) What are the characteristics of jointly encoding units in relation to model architectures and datasets? Training and implementation details can be found in the supplement.

4.1. What effect does model architecture have on static and dynamic biases?

4.1.1 Action recognition

Architectures. As the field of action recognition has largely moved away from explicit input motion representations (*e.g.* optical flow), we restrict our analysis to models that

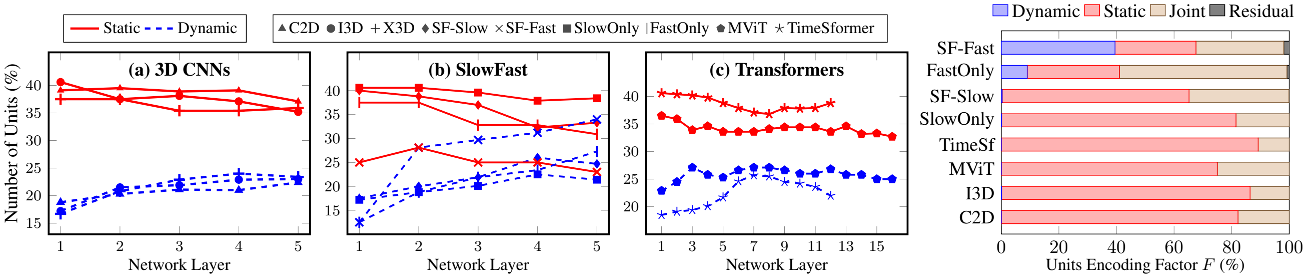

Figure 3. Layerwise and unit analyses on action recognition networks trained on Kinetics-400 [4]. **Left:** Layerwise encoding of static and dynamic factors using the layer-wise metric, (1), for: (a) single stream 3D CNNs, (b) SlowFast variants and (c) transformer variants. SF-Slow and SF-Fast denote the representation taken before the fusion layer from the slow and fast branches, resp. **Right:** Estimates of the dynamic, static, joint and residual units using the unit-wise metric, (3), on the final representation before the fully connected layer.

solely use the RGB modality. We study three types of models with respect to their static and dynamic biases: (i) single stream 3D CNNs (*i.e.* C2D [49], I3D [4] and X3D [13] models), (ii) SlowFast [14] variations, where we also study the two streams when trained individually, referred to as the SlowOnly and FastOnly models and (iii) transformer-based architectures [2, 12]. All models in this subsection are trained on the Kinetics-400 dataset [4] and taken from the SlowFast repository [14] without any training on our part (except FastOnly, which we implement). For all models, the number of frames and sampling rate is (8 × 8), except for the FastOnly network (32 × 2), MViT (16 × 4) and TimeSformer (8 × 32). To identify the static and dynamic units of all models, we generate the Stylized ActivityNet [11] validation set and use it for sampling *static* and *dynamic* pairs. We choose this dataset since the action distribution is similar to Kinetics-400, yet much smaller in size making it memory efficient when computing (1) and (3).

Layer-wise analysis. The static and dynamic units of multiple spatiotemporal models are quantified in Fig. 3 (left) using our layer-wise metric, (1). While the transformers are measured at every layer, the convolutional architectures are measured at five 'stages', corresponding to ResNet-50-like blocks [23]. We begin our examination by comparing the last layer (*i.e.* stage five) of each model, as this representation contains the final information before the model output. Interestingly, all single stream networks other than the FastOnly model are heavily biased toward *static* information even though the video frames of the static pairs are *randomly shuffled*. This result demonstrates the heavy bias toward static feature representations in these models. In fact, most of the 3D CNNs (*e.g.* I3D and SlowOnly) have a similar percentage of dynamic units as the C2D network, suggesting that these models do not sufficiently capture complex dynamic representations.

We perform the static and dynamic estimation on the representations for the slow and the fast branch of the SlowFast model separately (*i.e.* before fusion of the features). As shown in Fig. 3 (b), this dual-stream technique for capturing dynamic information works well, as the fast branch has a significant number of dynamic units, even without the use of optical flow as input. Notably, this finding also holds for other datasets as well (see Sec. 4.2). One key component of the SlowFast network is the fusion branch that aims to transfer information from the fast branch to the slow branch. This operations is performed by concatenating the slow and fast features followed by a time-strided convolution. Since the SlowOnly network is simply the SlowFast network without the fast branch, comparing the dynamic and static between the SlowOnly and SlowFast (slow) branch can reveal whether dynamic information is transferred between the pathways. The addition of the fast pathway increases the dynamic units in the slow pathway by 3.3% as early as stage two. Additional experiments in the supplement show the robustness of our conclusion with a varying number of input frames and sampling rates.

Looking beyond solely the final layer of the models reveal a number of interesting observations. Fig. 3 demonstrates how all models are biased toward *static* information at the earlier layers, with a tendency to encode more dynamics deeper in the network. The C2D, I3D and X3D models have only small, generally monotonic, changes in dynamic and static information at each stage. The SlowFast-Fast branch has the largest change in terms of the dynamic units, again showing the ability of the two-stream architecture to capture dynamic information. Conversely, the per-layer characteristics of static and dynamic encoding is different in both transformer-based architectures. They encode an increasing amount of dynamic information up until about halfway through the model, at which point the pattern tapers off and even reverses slightly.

Unit-wise analysis. We now examine individual units using our unit-wise metric, (3), with $\lambda = 0.5$ and report the results for the final representation before the fully connected layer in Fig. 3 (right). Interestingly, all single stream models, other than FastOnly, contain mainly *static* and *joint* units. There appears to be no difference between single-stream transformers and CNNs in the emergence of dynamic or residual units. In contrast, the FastOnly model and SlowFast-Fast branch produce a significant number of

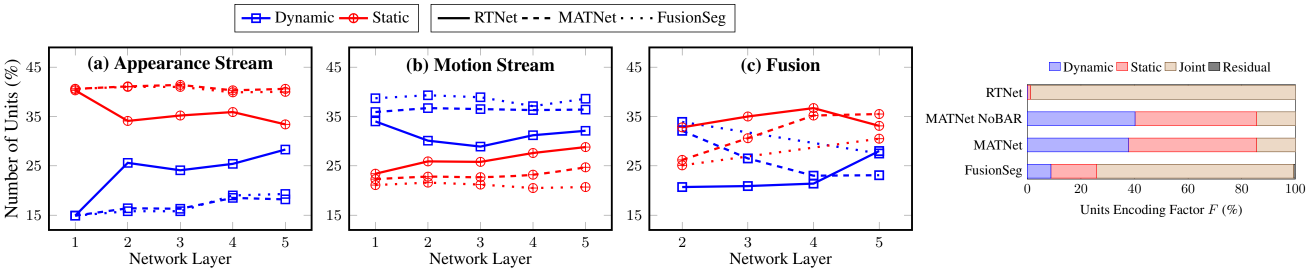

Figure 4. Layer and unit-wise analysis on off-the-shelf VOS networks. **Left**: Encoding of dynamic and static factors for motion, appearance streams and fusion layers in FusionSeg [28], MATNet [52] and RTNet [38] using the layer-wise metric, (1). Fusion layers are mostly biased towards the static factor. **Right**: Unit analysis for the three models targeting fusion layer five using the unit-wise metric, (3). MATNet has the largest number of dynamic units. MATNet NoBAR represents MATNet without the boundary-aware refinement module.

dynamic units. Another finding consistent with the results from Fig. 3 (right), is revealed when comparing the FastOnly model and SlowFast-Fast branch: The Fast model extracts more dynamic information *when trained jointly with the Slow branch*. These findings all together demonstrate the efficacy of two-stream architectures with varying capacity and frame rates. In the supplement, we verify that this pattern of results remain consistent while varying the threshold, λ, and provide results at multiple layers.

4.1.2 Video object segmentation

Architectures. We study the dynamic and static biases of two-stream fusion VOS models that take two-frame optical flow and an RGB image as input, with different types of cross connections: (i) FusionSeg [28] with no cross connections, (ii) MATNet [52] with motion-to-appearance cross connections and (iii) RTNet [38] with bidirectional cross connections. For a fair comparison with the two other models that fuse motion and appearance in the intermediate representations, we use a modified version of FusionSeg [28] trained on DAVIS16 [37] in our analysis. Our modified model follows an encoder-decoder approach [5], resulting in two fusion layers as detailed in the supplement. Our model achieves similar performance to the original on DAVIS16 (70.8% vs. 70.7% mIoU). For both MATNet [52] and RTNet [38], we use the models provided by the authors without further fine-tuning. We provide an analysis on MATNet trained only on DAVIS16 (*i.e.* without additional YouTube-VOS data) in the supplement. We use a stylized version of DAVIS16 in our analysis to evaluate the static and dynamic biases for the previous models, with stylization according to Sec. 3.1. In the case of both motion and appearance streams, we analyse features after cross connections, if present. In the case of fusion layers, the features extracted after the spatiotemporal attention fusion in RTNet, and the features after scale sensitive attention in MATNet are used. In FusionSeg, the features after the convolutional layers fusing motion and appearance from the second and fifth ResNet stages are used.

Layer-wise analysis. Figure 4 (left), shows the layerwise analysis for the motion and appearance streams as well as the fusion layers according to our layer-wise metric, (1). Similar to our finding with the action recognition models in Sec. 4.1.1, the majority of the video object segmentation models are biased toward the *static* factor in the fusion layers (*i.e.* fusion layers three, four and five). We observe an increase in the dynamic bias in the appearance stream as we go deeper in the network, especially for RTNet. In contrast, the bias in the motion streams of both FusionSeg and MATNet are somewhat consistent throughout layers. Interestingly, in RTNet, the *static* bias increases as the representation goes deeper in the network. This result likely stems from the bidirectional cross-connections in RTNet.

Unit-wise analysis. The individual unit analysis for these models obtained using our unit-wise metric, (3), with $\lambda = 0.5$ is shown in Fig. 4 (right) for fusion layer five. MATNet has a nontrivial increase of dynamics biased units compared to the other models. In contrast, RTNet and FusionSeg show a greater number of jointly encoding units, coming at the expense of units biased toward the static and dynamic factors. This pattern suggests that cross connections, as present in MATNet, can lead to an increase in the specialized units that encode the static and dynamic factors in the late fusion layers. We also show MATNet trained without its boundary-aware refinement module and boundary loss, as "MATNet NoBAR", confirming the source behind such an increase are the motion-to-appearance cross connections.

As with action recognition, experiments in the supplement demonstrates that our observations are robust with respect to different fusion layers, variations of the threshold, λ, and training dataset variations (*i.e.* without YouTube-VOS). In the supplement, we also demonstrate that motion-to-appearance cross connections relate to the performance for a task requiring dynamic information (*i.e.* the segmentation of camouflaged moving objects (MoCA) [31]).

4.1.3 Summary and shared insights

We have shown in both action recognition and video segmentation that the majority of the examined state-of-the-art models are biased toward encoding static information. We

Dataset	SlowOnly		FastOnly	
	Dyn.(%)	Stat.(%)	Dyn.(%)	Stat.(%)
Kinetics	21.4	38.4	27.3	30.9
Diving48	23.1	34.0	23.8	27.3
SSv2	28.2	30.7	31.6	21.9

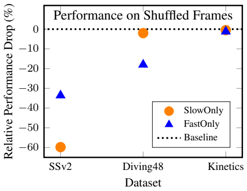

Figure 5. Analyses of biases of action recognition datasets. **Left:** *Dynamic* and *static* dimensions using the layer-wise metric, (1), for networks trained on Kinetics-400 [4], Diving48 [33] and SSv2 [19]. **Right:** Relative percentage drop in Top 1 Accuracy (%) for the SlowOnly and FastOnly models trained with shuffled frames with respect to the baseline (*i.e*. standard training). SSv2 drops more in performance than Diving48 or Kinetics-400.

also demonstrated the efficacy of two-stream models with motion-to-appearance [52] (fast-to-slow [14]) cross connections to enable greater encoding of dynamic information. Finally, we documented that the final layers of dynamic biased models are capable of producing a significant amount of specialized dynamic units compared to the joint units produced by static biased models.

4.2. How does the training dataset affect static and dynamic biases?

4.2.1 Action recognition

Datasets. With the knowledge that action recognition models often use static context biases in the data to make predictions (*e.g*. [6,8]), we consider datasets in the following evaluations which were designed with the goal of benchmarking a model's ability to capture dynamic information. Two popular datasets of this type are Something-Something-v2 [19] (SSv2) and Diving48 [33]. SSv2 is a fine-grained egocentric dataset with 174 classes and over 30,000 unique objects. Notably, different actions in SSv2 include similar appearance but different motions, *e.g*. the classes 'moving something from right-to-left' and 'moving something from left-to-right'. Diving48 [33] was created to be "a dataset with no significant biases toward static or short-term motion representations, so that the capability of models to capture long-term dynamics information could be evaluated" [32]. All actions are a particular type of dive and differ by only a single rotation or flip. We compare Kinetics-400, Diving48 and SSv2 to determine the extent that each dataset requires dynamics for action recognition.

Dataset bias. We use the layerwise metric, (1), to estimate the static and dynamic units captured in the last layer of two models trained on the three datasets, as shown in the table of Fig. 5 (left). We generate Stylized SSv2 and Stylized Diving48 to produce the static and dynamic estimates (and continue using Stylized ActivityNet for Kinetics-400 trained models). We measure the last layer, as the final prediction is made directly from it and thus is most representative of what information the model uses for the final prediction. The SlowOnly and FastOnly architectures follow a similar pattern to that found in Sec. 4.1, with the FastOnly consistently capturing more dynamic information. Surprisingly, models trained on Diving48 capture a similar amount of dynamics compared to Kinetics. These results may seem curious at first, as it seems unlikely that models could perform well on Diving48 without dynamic information.

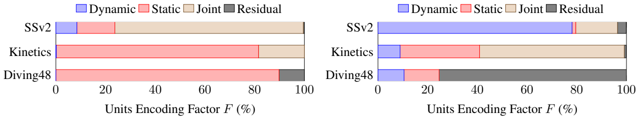

Figure 6. Estimating the dynamic, static, joint, and residual units using the unit-wise metric, (3), for the SlowOnly (**left**) and FastOnly (**right**) models on Kinetics-400 [4], Diving48 [33] and SSv2 [19]. Dynamic units arise from dynamic-biased models (*e.g*. FastOnly) and residual units from training on Diving48.

To further understand and confirm this result, we conduct a simple experiment, where the model only has static information to learn from. As discussed in Sec. 3.1, frame-shuffled videos will have the same static information as a non-shuffled input, but the temporal correlations, and hence dynamic information, will be corrupted. This manipulation forces the model to focus on static information for classification. We compare the top-1 validation accuracy of models trained and validated on shuffled frames to that of models with standard training. Fig. 5 (right) shows the results of the SlowOnly and FastOnly networks on Diving48, SSv2 and Kinetics-400, in terms of the relative performance on shuffled frames compared to unshuffled. For a fair comparison, we initialize all models from Kinetics-400. Both models show strong relative performance when trained to classify shuffled videos for Diving48 and Kinetics-400; however, for SSv2 the classification performance is decreased to a greater extent when trained on shuffled frames. These results show that SSv2 is a better alternative for benchmarking temporally capable networks.

Individual units analysis. Figure 6 shows the individual units (from the last layer) for two models (one static biased, SlowOnly, and one dynamic biased, FastOnly) on Kinetics-400, Diving48 and SSv2. The SlowOnly model trained on Kinetics-400 contains only static and joint units. However, when trained on Diving48 or SSv2, both residual and dynamic units emerge, demonstrating the impact of the training dataset on producing specialized units. This finding is consistent across all static biased architectures; see supplement. Unlike the SlowOnly model, the FastOnly model contains many dynamic units trained on any dataset, showing the efficacy of the architecture for producing specialized dynamic units. Interestingly, each dataset is unique in the type of units that emerge. Diving48 produces residual units, suggesting there are other factors at play beyond dynamic and static information. On the other hand, SSv2 produces

Dataset	Fusion Layer 5		Fusion Layer 2	
	Dyn.(%)	Stat.(%)	Dyn.(%)	Stat.(%)
DAVIS	27.8	30.1	34.0	25.9
ImageNetVID	26.4	33.1	33.0	24.6
TAO-VOS	26.4	25.8	33.7	23.2

Table 1. Biases of video object segmentation datasets using the layer-wise metric, (1), for FusionSeg's fusion layers five and two, trained on DAVIS16 [37], ImageNetVID [28] and TAO-VOS [46].

the most dynamic units for both models. The supplement shows this observation is consistent with other models.

4.2.2 Video object segmentation

Datasets. We study the impact of the following three VOS datasets on a model's static and dynamic biases: DAVIS16 [37], Weakly Labelled ImageNet VID [28] and TAO-VOS [46]. DAVIS16 [37] is the most widely used benchmark for automatic VOS, with 50 short-temporal extent sequences of two to four seconds and 3455 manually annotated frames. ImageNet VID [28] contains 3251 weakly labelled videos and was used in previous work to pretrain a model's motion stream [28]. Here, we use it as a general training dataset, *i.e.* beyond just for motion streams, to assess its impact. Finally, TAO-VOS [46] contains 626 relatively long videos (36 seconds on average) that are annotated in a hybrid fashion between manually and weakly labelled frames, resulting in 74,187 frames. We convert the annotations to exclude instances and instead consider foreground/background annotations only.

Dataset bias. We train our modified version of FusionSeg with early (layer two) and late (layer five) fusion layers on our three datasets. We compute the static and dynamic biases for the training datasets using the layer-wise metric, (1), and report the results in Table 1. The model trained on TAO-VOS has the least amount of static bias out of all three datasets. However, it appears that the datasets do not differ significantly in their dynamic bias. These results are further explored, by analyzing the specialized dynamic and jointly encoding units, as discussed in the next section.

Individual units analysis. We analyse the datasets in terms of the individual unit analysis using the unit-wise metric, (3), with $\lambda = 0.5$. It is seen in Fig. 7 (left) that models trained on TAO-VOS produce the highest number of specialized dynamic biased units, unlike DAVIS16 and ImageNet VID that show more joint units. To explore this matter further, we evaluate the center bias for the three datasets by calculating the average (normalized to 0-1) number of groundtruth segmentation masks for each pixel over the entire dataset, with results shown in Fig. 7 (right). It is seen that for both layers, the percentage of specialized dynamic units is greatest for the dataset that has least center bias, *i.e.* TAO-VOS, as its center bias map is far more diffuse than the others. These observations have implications for how the datasets can be used best for different tasks. For example, more general motion segmentation without concern for centering, might be better served by training with a dynamic biased dataset (*e.g.* TAO-VOS) unlike static biased datasets (*e.g.* DAVIS16 and ImageNet VID).

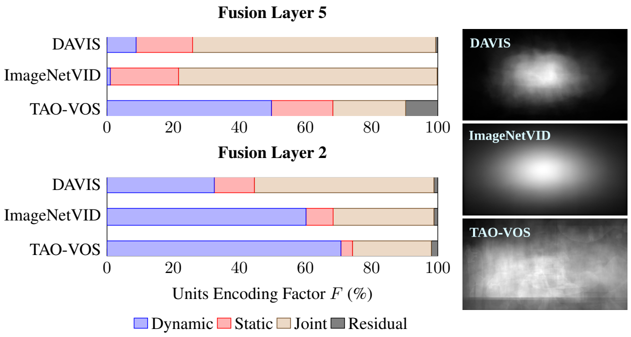

Figure 7. Analyses of biases of VOS datasets. **Left**: Estimating the dynamic, static, joint and residual units using the unit-wise metric, (3), for FusionSeg's fusion layers five and two trained on DAVIS16 [37], ImageNetVID [28] and TAO-VOS [46]. **Right**: Center bias plots for the three datasets. The results show the emergence of more dynamic units for both fusion layers when trained on the least center biased dataset (*i.e.* TAO-VOS).

4.2.3 Summary and shared insights

We have shown the effect of training datasets on both tasks. Our results raise questions about some of the widely adopted datasets in action recognition. In particular, Diving48 is claimed to be a good benchmark for learning dynamics [33]. Instead, our results suggest that SSv2 is better suited for evaluating a model's ability to capture dynamics. In video object segmentation, we found training on TAO-VOS yields the largest number of specialized dynamic units. Thus, it may be a better training dataset for tasks that rely on capturing dynamics (*e.g.* motion segmentation).

5. Conclusion

This paper has advanced the understandability of learned spatiotemporal models for video understanding, especially action recognition and video object segmentation. We have introduced a general method for analyzing the extent that various architectures capitalize on static vs. dynamic information. We also showed how our method can be applied to investigate the static vs. dynamic biases in datasets. Future work can apply our method to additional video understanding tasks (*e.g.* action prediction) as well as use insights gained on particular models and datasets to improve their performance and applicability (*e.g.* reduce identified biases for better generalization to new data).

Acknowledgements. We gratefully acknowledge financial support from the Canadian NSERC Discovery Grants and Vector Institute Post-graduate Affiliation award. K.G.D. and R.P.W. contributed to this work in their personal capacity as Associate Professors at York University.

References

[1] Nayyer Aafaq, Ajmal Mian, Wei Liu, Syed Zulqarnain Gilani, and Mubarak Shah. Video description: A survey of methods, datasets, and evaluation metrics. *ACM Computing Surveys*, 52(6):1–37, 2019. 2

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning*, 2021. 2, 5

[3] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Proceedings of the Conference on Fairness, Accountability and Transparency*, pages 77–91, 2018. 1

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 2, 5, 7

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 801–818, 2018. 6

[6] Jinwoo Choi, Chen Gao, C. E. Joseph Messou, and Jia-Bin Huang. Why can't I dance in the mall? Learning to mitigate scene bias in action recognition. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, 2019. 2, 7

[7] Achal Dave, Pavel Tokmakov, and Deva Ramanan. Towards segmenting anything that moves. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2

[8] Konstantinos G. Derpanis, Mikhail Sizintsev, Kevin J. Cannons, and Richard P. Wildes. Action spotting and recognition based on a spatiotemporal orientation analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):527–540, 2012. 7

[9] Konstantinos G. Derpanis and Richard P. Wildes. Spacetime texture representation and recognition based on a spatiotemporal orientation analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1193–1205, 2011. 3

[10] Patrick Esser, Robin Rombach, and Bjorn Ommer. A disentangling invertible interpretation network for explaining latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9223–9232, 2020. 2, 4

[11] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. 5

[12] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *arXiv preprint arXiv:2104.11227*, 2021. 2, 5

[13] Christoph Feichtenhofer. X3D: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020. 5

[14] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019. 2, 5, 7

[15] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. Spatiotemporal multiplier networks for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4768–4777, 2017. 2

[16] Christoph Feichtenhofer, Axel Pinz, Richard P Wildes, and Andrew Zisserman. Deep insights into convolutional networks for video recognition. *International Journal of Computer Vision*, 128(2):420–437, 2020. 2

[17] David V Foster and Peter Grassberger. Lower bounds on mutual information. *Physical Review E*, 83(1):010101, 2011. 4

[18] Amir Ghodrati, Efstratios Gavves, and Cees G. M. Snoek. Video time: Properties, encoders and evaluation. In *Proceedings of the British Machine Vision Conference*, 2018. 2

[19] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5842–5850, 2017. 2, 7

[20] Isma Hadji and Richard P Wildes. A new large scale dynamic texture dataset with application to convnet understanding. In *Proceedings of the European Conference on Computer Vision*, pages 320–335, 2018. 2

[21] Sven Ove Hansson, Matts-Åke Belin, and Björn Lundgren. Self-driving vehicles-An ethical overview. *Philosophy & Technology*, pages 1–26, 2021. 1

[22] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3D residual networks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 3154–3160, 2017. 2

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 5

[24] Yun He, Soma Shirakabe, Yutaka Satoh, and Hirokatsu Kataoka. Human action recognition without human. In *Proceedings of the European Conference on Computer Vision*, pages 11–17, 2016. 2

[25] Liam Hiley, Alun Preece, and Yulia Hicks. Explainable deep learning for video recognition tasks: A framework & recommendations. *arXiv preprint arXiv:1909.05667*, 2019. 2

[26] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017. 3

[27] Md Amirul Islam, Matthew Kowal, Patrick Esser, Sen Jia, Björn Ommer, Konstantinos G. Derpanis, and Neil D. B.

Bruce. Shape or texture: Understanding discriminative features in CNNs. In *Proceedings of the International Conference on Learning Representations*, 2021. 2, 4

[28] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. FusionSeg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2126. IEEE, 2017. 2, 3, 6, 8

[29] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2012. 2

[30] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004. 4

[31] Hala Lamdouar, Charig Yang, Weidi Xie, and Andrew Zisserman. Betrayed by motion: Camouflaged object discovery via motion segmentation. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 6

[32] Yingwei Li, Yi Li, and Nuno Vasconcelos. Diving48 dataset. `http://www.svcl.ucsd.edu/projects/resound/dataset.html`. Accessed: 2021-11-13. 7

[33] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision*, pages 513–528, 2018. 2, 7, 8

[34] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9572–9581, 2019. 2

[35] Joonatan Manttari, Sofia Broomé, John Folkesson, and Hedvig Kjellstrom. Interpreting video features: A comparison of 3D convolutional networks and convolutional LSTM networks. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 2

[36] Mandela Patrick, Dylan Campbell, Yuki M. Asano, Ishan Misra Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and Joao F. Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, 2021. 2

[37] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 6, 8

[38] Sucheng Ren, Wenxi Liu, Yongtuo Liu, Haoxin Chen, Guoqiang Han, and Shengfeng He. Reciprocal transformations for unsupervised video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15455–15464, 2021. 2, 3, 6

[39] 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 *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 535–544, 2021. 2

[40] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, volume 27, 2014. 2

[41] Graham W Taylor, Rob Fergus, Yann LeCun, and Christoph Bregler. Convolutional learning of spatio-temporal features. In *Proceedings of the European Conference on Computer Vision*, pages 140–153, 2010. 2

[42] Ondřej Texler, David Futschik, Michal Kučera, Ondřej Jamriška, Šárka Sochorová, Menclei Chai, Sergey Tulyakov, and Daniel Sỳkora. Interactive video stylization using few-shot patch-based training. *ACM Transactions on Graphics (TOG)*, 39(4):73–1, 2020. 3

[43] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4481–4490, 2017. 2

[44] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015. 2

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 2

[46] Paul Voigtlaender, Lishu Luo, Chun Yuan, Yong Jiang, and Bastian Leibe. Reducing the annotation effort for video object segmentation datasets. In *Proceedings of the IEEE Winter Conference on Computer Vision Applications*, 2021. 8

[47] Tuan-Hung Vu, Catherine Olsson, Ivan Laptev, Aude Oliva, and Josef Sivic. Predicting actions from static scenes. In *Proceedings of the European Conference on Computer Vision*, pages 421–436, 2014. 2

[48] Wenguan Wang, Tianfei Zhou, Fatih Porikli, David Crandall, and Luc Van Gool. A survey on deep learning technique for video segmentation. *arXiv preprint arXiv:2107.01153*, 2021. 1, 2

[49] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 5

[50] Richard P Wildes and James R Bergen. Qualitative spatiotemporal analysis using an oriented energy representation. In *European Conference on Computer Vision*, pages 768–784. Springer, 2000. 3

[51] He Zhao and Richard P Wildes. Interpretable deep feature propagation for early action recognition. *arXiv preprint arXiv:2107.05122*, 2021. 2

[52] Tianfei Zhou, Shunzhou Wang, Yi Zhou, Yazhou Yao, Jianwu Li, and Ling Shao. Motion-attentive transition for zero-shot video object segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13066–13073, 2020. 2, 3, 6, 7

[53] Yi Zhu, Xinyu Li, Chunhui Liu, Mohammadreza Zolfaghari, Yuanjun Xiong, Chongruo Wu, Zhi Zhang, Joseph Tighe, R

Manmatha, and Mu Li. A comprehensive study of deep video action recognition. *arXiv preprint arXiv:2012.06567*, 2020. 1, 2