Improving Video Model Transfer with Dynamic Representation Learning

Yi Li  Nuno Vasconcelos
Department of Electrical and Computer Engineering
University of California, San Diego

Abstract

Temporal modeling is an essential element in video understanding. While deep convolution-based architectures have been successful at solving large-scale video recognition datasets, recent work has pointed out that they are biased towards modeling short-range relations, often failing to capture long-term temporal structures in the videos, leading to poor transfer and generalization to new datasets. In this work, the problem of dynamic representation learning (DRL) is studied. We propose dynamic score, a measure of video dynamic modeling that describes the additional amount of information learned by a video network that cannot be captured by pure spatial student through knowledge distillation. DRL is then formulated as an adversarial learning problem between the video and spatial models, with the objective of maximizing the dynamic score of learned spatiotemporal classifier. The quality of learned video representations are evaluated on a diverse set of transfer learning problems concerning many-shot and few-shot action classification. Experimental results show that models learned with DRL outperform baselines in dynamic modeling, demonstrating higher transferability and generalization capacity to novel domains and tasks.

1. Introduction

Following the success of deep learning for image recognition [38, 48, 71, 75], convolutional neural networks have also gained popularity for video classification problems, such as action recognition [13, 25, 55, 70, 79, 84, 88], where they outperform other classification architectures, such as recurrent networks [20, 41, 89]. However, current action recognition performance is significantly below the levels of image recognition. This is, in part, due to the difficulty of learning video representations that generalize well. Part of this difficulty stems from current data collection practices, namely the use of action datasets collected from the web (e.g. YouTube) [10, 35, 45, 46, 59]. While these datasets have much larger size and diversity of action classes, performers and scenes than those collected in controlled settings [6, 66, 86], they are also known to exhibit various types of biases that hinder the generalization of trained video models to unseen domains [17, 53, 54, 67]. One of the most prevalent among these biases is the spatial bias due to the spurious correlation between action labels and the spatial appearance of video frames, in the form of background objects, scenes or human pose [53]. For example, the presence of a horse in the video is enough to infer the “horseback riding” action if that is the only action class in the dataset that involves horses. Spatial bias creates shortcuts that allow classifiers to infer action labels without modeling the temporal video component, known as video dynamics, leading to overoptimistic performance in popular action recognition benchmarks [54].

One of the nefarious consequences of dataset bias is that it favours certain representations over others [53]. In this context, the spatial bias of most video datasets is likely responsible for the dominant performance of convolutional architectures, known to favor local over long-range dependencies, in the action recognition field. This type of “evolutionary adaptation” of networks to dataset bias has, in fact, been documented in the object recognition literature, leading to the prevalence of networks with a strong bias towards local textures over global object shape [3, 8, 31]. We hypothesize that, in video modeling, the same phenomenon justifies the supremacy of 3D convolutional networks that implement localized spatio-temporal video representations, based on a very small number of frames, basically ignoring long-range video dynamics. This, we claim, hampers the generalization of these networks to unseen domains. For example, a horse representation is insufficient for transfer to a new domain that requires discrimination between “riding a horse” and “chasing a horse” or the classification of video into Olympic equestrian event classes. However, this problem is hard to diagnose in the predominant action recognition setting, where training and test data come from the same domain. If “horseback riding” is the only horse related class, detecting horses is enough for high performance on test data. Diagnosing the problem requires deploying the action recognizer outside of the native test set, as is usually done for tasks like few-shot learning [27, 64, 72, 74], domain adapta-
DRL alternates between two steps: Dynamic scoring step quantitatively evaluates the temporal dynamics captured by a video model, defined as the expected difference between its predictions and those of a spatial student; representation learning step updates parameters of the video network to optimize dynamic score.

We hypothesize that, when pre-trained on the same dataset with a given architecture, models of larger dynamic score should transfer and generalize better to unseen video domains. To evaluate this hypothesis, we conduct a set of experimental evaluations on the robustness and transferability of the learned video representation. This consists of a) adapting the video network to a set of datasets with different action vocabulary, using linear classification over learned features or fine-tuning; b) few-shot action classification using learned representations directly; and c) applying the classifier on a set of video actions in absence of their spatial context. Experimental results show that DRL significantly improves all three tasks. For example, 5-shot gesture recognition on the Jester dataset [58] is improved by 5% using 3D ResNet architecture [37], and up to 8% with TSM network [55].

The contributions of this paper is three-fold: First, we propose dynamic score, a quantitative measure of temporal modeling capacity of video neural networks. Second, we propose dynamic representation learning (DRL), a pre-training strategy that aims to optimize dynamic scores for video models. Finally, we present a comprehensive set of experiments to measure the transferability and generalization of learned video representations, which empirically validates the importance of dynamic modeling on video transfer learning and demonstrates the advantage of DRL over baseline pre-training methods.

2. Related Work

Deep video architectures. Following the success of deep neural networks for image recognition, convolutional architectures have dominated video action recognition. While some video neural networks use 2D spatial convolutions similar to those of image CNNs [26, 55, 70, 84], others rely on 3D convolutions operating in the spatiotemporal dimensions [13, 24, 37, 44, 79], or factorize the 3D convolution into a 2D spatial and a 1D temporal convolution [63, 80, 88]. An alternative to convolutions is to use

Figure 1. We propose dynamic representation learning (DRL), an adversarial learning strategy to enhance the modeling of video dynamics. DRL alternates between two steps: Dynamic scoring step quantitatively evaluates the temporal dynamics captured by a video model, defined as the expected difference between its predictions and those of a spatial student; representation learning step updates parameters of the video network to optimize dynamic score.

In this work, we address this problem by introducing a new approach to dynamic representation learning (DRL) based on the explicit measurement and minimization of spatial bias. As shown in Figure 1, DRL is based on an adversarial optimization between the video network and a spatial student, i.e., a 2D network that processes video frames independently. Video network and spatial student are optimized alternately. In a dynamic scoring step, shown in the left of the figure, the student is optimized to mimic the predictions of the video network. The expected difference between the predictions of the two networks reflects how dynamic the video representation is. This expected difference is denoted as the dynamic score of the video network, measuring how much it relies on dynamic, over spatial, cues for classification. The lower this score, the more similar the video model is to a spatial classifier, and the greater its spatial bias. While the dynamic score is measured naturally by training the student by knowledge distillation [40], we also propose an optimization-free approach based on the preprocessing of video clips to remove temporal information, which is more computationally efficient. In the representation learning step, shown on the right of Figure 1, the video model trained to maximize its dynamic score with respect to a learned spatial student. Two approaches are proposed for this purpose. The first is based on adversarial augmentation, using the spatial student to derive perturbations that, when added to the video, obscure spatial cues. The second poses DRL as a min-max game between the video network and the spatial student to directly optimize the dynamic score of the former. The two methods share the same key idea—to penalize the video model for leveraging spatial shortcuts to action recognition.
recurrent modules, such as the Long Short-term Memory (LSTM) [41], to model video dynamics [20, 89]. Attention mechanisms have also been studied to overcome the tendency of convolutional neural networks to favor short-range correlations over long-term dependencies. These include pooling convolutional features with self-attention [16, 85], or in more recent works [1, 5, 23], replacing all convolutional layers with Transformer blocks [21, 83].

**Dataset bias.** Computer vision datasets are known to exhibit biases, in that their image composition does not accurately resemble the real-world data distributions [47, 76, 77, 82]. In the context of video action recognition, Sigurdsson et al. [69] identified the domain gap between videos of human activities retrieved from the internet and our everyday actions. Li et al. [53] showed that many internet retrieved datasets have representation bias, favoring representations that capture spurious correlations between action labels and contextual cues, such as objects or scenes [46, 49, 73]. New datasets have been collected to overcome this limitation: Charades [69] and VLOG [28] used videos of daily activities, while Diving48 [53] and FineGym [68] considered sports domains rich in fine-grained action categories.

**Model bias.** Various forms of algorithmic bias have been discovered in a wide range of machine learning tasks. For example, studies have found that gender and racial bias in datasets can be exploited and sometimes amplified by machine learning models [7, 9, 39, 91]. In image recognition and object detection, contextual bias from background objects or scenes have been shown to encourage the learning of models with poor generalization performance on novel objects or scenes [46, 49, 73]. New datasets have been collected to overcome this limitation: Charades [69] and VLOG [28] used videos of daily activities, while Diving48 [53] and FineGym [68] considered sports domains rich in fine-grained action categories.

**Knowledge distillation.** First introduced by Hinton et al. [40], knowledge distillation is a method to transfer information from a teacher model to a (usually weaker) student model. Initially introduced as a solution to model compression, the technique has since been adopted to other problems, including adversarial defenses [62] and cross-modal transfer [36].

### 3. Dynamic Scoring

In this section, we introduce the dynamic score, a measure of how a representation captures video dynamics.

#### 3.1. Definition

A video representation is a mapping $\phi : \mathcal{X} \rightarrow \mathcal{Z}$ from some space $\mathcal{X}$ of videos to a feature space $\mathcal{Z}$. A $K$-way video classifier is the mapping $f_{\phi} = h \circ \phi$ composed by a feature representation $\phi$ and a linear classifier $h : \mathcal{Z} \rightarrow \mathbb{S}^K$, where $\mathbb{S}^K$ is the $K$-dimensional probability simplex. For a generic video classifier that processes video clips of $T$ frames of dimension $D$, $\mathcal{X} = \mathbb{R}^{T \times D}$. The video classifier is denoted as purely spatial if it applies a spatial representation $\phi_s$ to video frames independently, i.e. if

$$f_{\phi_s} = \frac{1}{T} \sum_{t=1}^{T} g_{\phi_s}(x_t)$$

for some image classifier $g_{\phi_s}$. Let $\mathcal{F}^S$ be the set of all such classifiers and $\mathcal{L} : \mathbb{S}^K \times \mathbb{S}^K \rightarrow [0, \infty)$ a similarity score for model predictions. The dynamic score of model $f_{\phi}$ with respect to a distribution $p_D$ of video clips is then defined as

$$\gamma(f_{\phi}; p_D) = \min_{f_{\phi_s} \in \mathcal{F}^S} \mathbb{E}_{x \sim p_D} \mathcal{L}(f_{\phi_s}(x), f_{\phi}(x)).$$

This is zero when $f_{\phi}$ is purely spatial ($f_{\phi} \in \mathcal{F}^S$) and increases with the ability of the representation to capture video dynamics, i.e. temporal features of the video. While the definition above supports any similarity score between probability distributions, we use the Kullback–Leibler (KL)
In this case, $\gamma(f_\phi; p_D)$ can be intuitively interpreted as the amount of dynamic information captured by $\phi$. The definition of dynamic score can also be easily generalized beyond classification problems by application of an appropriate similarity metric $\mathcal{L}$.

### 3.2. Measuring the dynamic score

The empirical measurement of $\gamma(f_\phi; p_D)$ requires finding the spatial model $f_{\phi_s}$ of minimum disagreement with $f_\phi$, as measured by $\mathcal{L}(\cdot)$. Since it is impractical to search the entire space $\mathcal{F}^S$, we consider a few strategies to find a near-optimal $f_{\phi_s}$ efficiently.

**Knowledge distillation.** If the search space $\mathcal{F}^S$ is constrained to deep neural networks of a specific architecture, equation 2 reduces to knowledge distillation [40] from the video teacher model $f_\phi$ to the spatial student classifier $f_{\phi_s}$. As illustrated in figure 2a, the student model is trained to predict the outputs of the video network $f_\phi(x)$; and $\gamma(f_\phi; p_D)$ is the distillation loss on the test set. We use spatial models of the form of equation 1, using a standard 2D convolutional neural network as function $g$.

**Input modulation.** Knowledge distillation requires training a student network $f_{\phi_s}$ different from $f_\phi$ from scratch. Alternatively, we consider a training free procedure, where the spatial model $f_{\phi_s}$ is derived from the network $f_\phi$ itself, by pre-processing its input video clips in a way that removes temporal information. As shown in figure 2b, this can be achieved by either “freezing” the clip, i.e. sampling a single frame and repeating it along the temporal dimension, or by reshuffling the frames in a random order. With $\tilde{x}$ denoting the frozen or re-shuffled clip, dynamic score is approximated by

$$\gamma_\alpha(f_\phi; p_D) = \mathbb{E}_{x \sim p_D} \mathcal{L}(f_{\phi_s}(\tilde{x}), f_\phi(x)).$$

### 3.3. Relation to dataset bias

While the dynamic score is a measure for video representations, it is closely related to prior measures of the spatial bias of video action datasets, which translates into unexpectedly high action recognition performance by purely spatial classifiers [43, 53]. By replacing the video model output $f_\phi(x)$ in equation 2 with the ground-truth label $y$, the formula becomes a measure of dataset bias,

$$\gamma(p_D) = \min_{f_{\phi_s} \in \mathcal{F}^S} \mathbb{E}_{x,y \sim p_D} \mathcal{L}(f_{\phi_s}(x), y),$$

quantifying the difficulty posed by dataset $\mathcal{D}$ to purely spatial recognition models. Analogous to the dynamic score, a higher score $\gamma(p_D)$ indicates that more temporal modeling is required to correctly classify videos of $\mathcal{D}$, as even the optimal spatial classifier performs poorly on the dataset. Importantly, this implies that the dynamic score $\gamma(f_\phi; p_D)$ also reflects the static bias of the data. If the videos of $\mathcal{D}$ comprise mostly spatial cues (i.e. $\mathcal{D}$ has a large static bias), model predictions $f_\phi(x)$ can be easily fitted by a purely spatial model $f_{\phi_s}$, resulting in lower $\gamma(f_\phi; p_D)$. In fact, the dynamic score of the oracle classifier $f^*$, which predicts action class $y$ with 100% accuracy, is equivalent to the dataset bias, i.e. $\gamma(f^*, p_D) = \gamma(p_D)$.

### 4. Dynamic Representation Learning

In this section we discuss two approaches to dynamic representation learning (DRL). DRL by data augmentation applies adversarial perturbations to the input data to increase the difficulty of pure spatial modeling. DRL by direct optimization solves a min-max problem involving the parameters of the video and spatial networks.

#### 4.1. DRL by augmentation

Inspired by the success of adversarial training to improve model robustness [33, 50, 57, 78], this DRL approach is based on the generation of training samples adversarial to the spatial model $f_{\phi_s}$. This reduces the effectiveness of spatial modeling, forcing the video network $f_\phi$ to model video dynamics. Adversarial perturbations are generated by any
of the techniques in the adversarial attack literature, such as FGSM [33] or PGD [57]. The perturbation is adversarial to the purely spatial model \( f_{\phi_s} \), so as to weaken any spatial cues for action recognition. For example, for FGSM, perturbations are generated with

\[
\tilde{x} = x + \epsilon \operatorname{sgn} \left[ \nabla_x \mathcal{L}_{kd}(f_{\phi_s}(z), f_{\phi}(x)) \right]_{z=x},
\]

where \( \epsilon \) controls the size of the adversarial perturbation and the gradient is only backpropagated through \( f_{\phi_s} \). As usual for adversarial training, the video model is then trained with a combination of classifications losses on the original and perturbed data

\[
\min_{f_{\phi} \in \mathcal{F}} \mathbb{E}_{x,y \sim p_{D}} \left[ \beta \mathcal{L}_{cls}(f_{\phi}(\tilde{x}), y) + (1 - \beta) \mathcal{L}_{cls}(f_{\phi}(x), y) \right],
\]

where \( \beta \) is an hyperparameter. To maximize training efficiency, the video \( f_{\phi} \) and spatial \( f_{\phi_s} \) models are trained jointly, using algorithm 1. At each step, \( f_{\phi_s} \) is first trained to mimic the predictions of \( f_{\phi} \) by distillation. As usual, this includes a combination of a class label prediction and a KL loss, weighted by a factor \( \alpha \). The adversarial example \( \tilde{x} \) is then generated against \( f_{\phi_s} \) and \( f_{\phi} \) is finally updated. The whole procedure can be seen as a defense mechanism that forces \( f_{\phi} \) to learn a dynamic representation.

### 4.2. DRL by direct optimization

This approach maximizes the dynamic score of the video network directly during training. This translates to a min-max game between video model (teacher) and spatial model

<table>
<thead>
<tr>
<th>Algorithm 2: DRL iteration by min-max optimization.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Mini-batch ( B \subset D ), video model ( f_{\phi} ) w/ parameters ( \theta ), spatial model ( f_{\phi_s} ) w/ parameters ( \psi ); learning rate ( \eta ), distillation weight ( \alpha ), dynamic loss weight ( \lambda )</td>
</tr>
<tr>
<td><strong>for</strong> ((x, y) \in B) <strong>do</strong></td>
</tr>
<tr>
<td>Update spatial model by knowledge distillation</td>
</tr>
<tr>
<td>( \psi \leftarrow \psi - \eta \nabla_{\psi} \left[ \alpha \mathcal{L}<em>{kd}(f</em>{\phi_s}(x; \psi), f_{\phi}(x; \theta)) + (1 - \alpha) \mathcal{L}<em>{cls}(f</em>{\phi_s}(x; \psi), y) \right] )</td>
</tr>
<tr>
<td>Update video model to maximize dynamic score</td>
</tr>
<tr>
<td>( \theta \leftarrow \theta - \eta \nabla_{\theta} \left[ \mathcal{L}<em>{cls}(f</em>{\phi_s}(x; \psi), f_{\phi}(x; \theta)) - \lambda \cdot \mathcal{L}<em>{kd}(f</em>{\phi_s}(x; \psi), f_{\phi}(x; \theta)) \right] )</td>
</tr>
<tr>
<td><strong>end</strong></td>
</tr>
<tr>
<td><strong>Output:</strong> Updated model parameters ((\theta, \psi))</td>
</tr>
</tbody>
</table>

As described in algorithm 2, we adopt a training procedure similar to that of adversarial networks [29, 30, 32], in this case alternating between updating the parameters of \( f_{\phi} \) and \( f_{\phi_s} \). The maximization over \( f_{\phi_s} \) optimizes the spatial model \( f_{\phi_s} \) to mimic the predictions of the video model \( f_{\phi} \) as closely as possible. The minimization over \( f_{\phi} \) then forces the latter to produce predictions that are as different as possible from those of the former, while minimizing classification error, leading \( f_{\phi} \) to learn video dynamics. This adversarial objective has a similar form to the ReBias algorithm [2]. The two methods differ primarily in the similarity criteria between spatial and temporal models: While [2] optimized the Hilbert-Schmidt Independence Criterion (HSIC) between feature spaces learned by the de-biased and biased models, DRL uses the distillation loss, i.e. KL divergence, between their output probabilities. We believe that this allows the proposed dynamic score to explicitly capture the agreement between the spatial and dynamic models in classifying the action categories.

Similarly to the score computation of section 3.2, the distillation of the static model can be replaced by a training free approach, where \( f_{\phi_s} \) is derived from \( f_{\phi} \) itself by pre-processing the input video to remove temporal information. In this case, algorithm 3 is used.

### 5. Experiments

In this section, we perform dynamic scoring of existing video recognition networks and evaluate how DRL improves their ability to model video dynamics.
5.1. Experimental setup

**Datasets.** We adopt the practice of pre-training video recognition models on the Kinetics-400 [46] dataset. For preliminary dynamic scoring experiments (sect. 5.2), the models are scored on a 200-class subset of Kinetics, miniKinetics [88], to reduce training time. As summarized in Table 1, the impact of DRL on model transfer (sect. 5.3) is evaluated in six datasets of different action domains. UCF-101 [73], HMDB [49], Diving-48 [53] are used to evaluate transfer by fine-tuning, while Something-Something v2 [34] and Jester [58] are used to evaluate few-shot classification. We do not test these two datasets for standard classification due to their large size, which diminishes the need for model pretraining. We also evaluate domain generalization of trained classifiers without any fine-tuning on the Mimetics [87] test set, which shares action vocabulary with Kinetics but has weaker spatial biases.

**Models.** DRL is evaluated on video action recognition networks with different types of convolutional modules: 3D ResNet [37] uses 3D convolution kernels, while TSM [55] is based on 2D convolutions. Sampling frame rate is adaptively chosen to ensure a constant length for input clips of 1 second, so that dynamic scores are comparable across models. Detailed training procedure included in supplemental.

### Table 1. Datasets for pre-training and evaluation. 5 classes are randomly drawn during each few-shot learning episode. Few-shot training sets consist of 5 (1-shot) or 25 (5-shot) examples. Here we report the total number of classes and samples of the full datasets.

<table>
<thead>
<tr>
<th>Training mode</th>
<th>Dataset</th>
<th># classes</th>
<th># examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-training</td>
<td>Kinetics [46]</td>
<td>400</td>
<td>300k</td>
</tr>
<tr>
<td></td>
<td>miniKinetics [88]</td>
<td>200</td>
<td>85k</td>
</tr>
<tr>
<td>Fine-tuning</td>
<td>UCF-101 [73]</td>
<td>101</td>
<td>13k</td>
</tr>
<tr>
<td></td>
<td>HMDB [49]</td>
<td>51</td>
<td>5k</td>
</tr>
<tr>
<td></td>
<td>Diving-48 [53]</td>
<td>48</td>
<td>18k</td>
</tr>
<tr>
<td>Few-shot</td>
<td>Sth-Sth V2 [34]</td>
<td>174</td>
<td>220k</td>
</tr>
<tr>
<td>recognition</td>
<td>Jester [58]</td>
<td>27</td>
<td>148k</td>
</tr>
<tr>
<td>Domain</td>
<td>Mimetics [87]</td>
<td>50</td>
<td>700</td>
</tr>
<tr>
<td>generalization</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.2. Dynamic scoring

Table 2 summarizes the dynamic scores of networks learned on miniKinetics [88]. The Baseline is a model trained by standard cross-entropy minimization. Dynamic scores are measured by either the distillation score of equation 2 (Distill), or the approximate score of equation 4, based on removing temporal information from input clips (Freeze and Shuffle). DRL is implemented with adversarial augmentation (algorithm 1), direct optimization (algorithm 2), or its approximation (algorithm 3) based on clip freezing or shuffling. Several conclusions can be drawn from the table. First, all DRL methods are effective at improving the dynamic score $\gamma$ of the baseline. The gains are larger for DRL by direct optimization, which frequently doubles the baseline score. Second, as usual for min-max games, the optimization of algorithm 2 is not easy. We found approximate DRL using frozen clips to be easier to train and, as shown in the table, lead to the best dynamic score. The same does not hold for approximate DRL with frame shuffling, which has the worst performance of the three direct optimization techniques. We believe that this is because frame shuffling creates artificially high temporal frequencies that the video network cannot model. Approximate DRL by direct optimization with frozen clips is used in all subsequent experiments. Third, regarding dynamic score measures, while different approaches produced different values of $\gamma$, the relative order of the different learning methods remained constant. While knowledge distillation provides the smallest (hence the most accurate) estimate of dynamic score, it requires training a new image convnet from scratch. Since measurement by freezing or shuffling input clips merely requires a single forward pass through the test set, it is much more efficient to evaluate.

### Table 2. Dynamic scores of 3D ResNet-18 models trained on miniKinetics-200, measured by distillation or freezing/shuffling clips. “Baseline” denotes standard cross-entropy training; DRL variants are discussed in sect. 4. Best results in **bold**, runners-up underlined.

<table>
<thead>
<tr>
<th>Method</th>
<th>Dynamic score $\gamma(f_{\phi}, P_D)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Baseline</td>
</tr>
<tr>
<td>DRL</td>
<td></td>
</tr>
<tr>
<td>(Adv. augment)</td>
<td></td>
</tr>
<tr>
<td>Distill, $\lambda = 0.5$</td>
<td></td>
</tr>
<tr>
<td>Freeze, $\lambda = 0.5$</td>
<td></td>
</tr>
<tr>
<td>Shuffle, $\lambda = 0.5$</td>
<td></td>
</tr>
</tbody>
</table>

5.3. Transfer learning

**Many-shot recognition.** Table 3 compares transfer learning performance from Kinetics to the UCF-101 [73], HMDB [49] and Diving-48 [53] datasets, using either linear evaluation over pretrained representations or full network fine-tuning. We compare the DRL models, trained by approximate optimization with frozen clips (algorithm 3), to baseline cross-entropy pre-training on Kinetics. Fine-tuning accuracies are also compared to networks initialized from scratch (3D ResNet) or ImageNet weights (TSM). We observe the following: First, transfer learning from Kinetics significantly outperforms training from scratch or with ImageNet initialization. This a well known consequence of the small target dataset sizes and confirms the importance of Kinetics pre-training for many action recognition datasets in the literature. Second, DRL is quite effective at increasing the dynamic score of all models, frequently doubling or even tripling the score of the baseline. Third, DRL achieves top performance in both the linear setting
and with fine-tuning for almost all combinations of model and target dataset. In general, the gains with fine-tuning are smaller than that for linear classification. This is expected, as network fine-tuning can correct some of the biases of the pre-trained model. This is especially true for UCF-101, which shares very similar biases with the Kinetics training set, as both are constructed from the same data source (YouTube), and consist of similar action classes. This leads to high accuracies under the transfer setting, and smaller improvements from DRL. The gains of DRL are consistent on HMDB and Diving, where video sources and action vocabularies diverge sufficiently from Kinetics. Overall, DRL improves linear classification accuracy by 2.39% on average, and fine-tuning accuracy by 1.03%.

We also observe that an increased dynamic score does not translate into accuracy gains on the Kinetics dataset, where DRL models underperform baseline pre-training by 2–3%. This is expected since the training and test set have the same spatial biases, such that models trained to exploit all biases achieve higher accuracy. Nonetheless, when evaluated under a distribution shift, these spatial models tend to transfer worse than models trained with DRL objective.

Table 3. Linear classification and fine-tuning performance of Kinetics models on UCF-101, HMDB-51 and Diving-48. By improving dynamic score \( \gamma(f_\phi, p_D) \), DRL produces transferable video representations, both for linear discrimination and fine-tuning.

<table>
<thead>
<tr>
<th>Method</th>
<th>Architecture input</th>
<th>Pretrain acc. K400</th>
<th>Linear acc. UCF</th>
<th>Fine-tuning acc. UCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scratch</td>
<td>3D ResNet-18</td>
<td>56.40</td>
<td>83.27</td>
<td>59.08</td>
</tr>
<tr>
<td>Baseline</td>
<td>112x112x16</td>
<td>53.32</td>
<td>52.29</td>
<td>61.24</td>
</tr>
<tr>
<td>DRL</td>
<td>112x112x16</td>
<td>84.30</td>
<td>11.32</td>
<td>63.59</td>
</tr>
<tr>
<td>Scratch</td>
<td>3D ResNet-50</td>
<td>62.04</td>
<td>87.23</td>
<td>46.55</td>
</tr>
<tr>
<td>Baseline</td>
<td>112x112x16</td>
<td>59.92</td>
<td>59.54</td>
<td>64.58</td>
</tr>
<tr>
<td>DRL</td>
<td>112x112x16</td>
<td>88.24</td>
<td>16.55</td>
<td>67.92</td>
</tr>
</tbody>
</table>

Table 4. Cross-domain few-shot evaluation of video representations on Something-Something V2 and Jester. Pre-training on Kinetics; 5-way accuracies reported in all experiments.

<table>
<thead>
<tr>
<th>Method</th>
<th>Architecture input</th>
<th>Sth-Sth v2 1-shot</th>
<th>Jester 1-shot</th>
<th>Jester 5-shot</th>
<th>Jester 5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scratch</td>
<td>3D ResNet-18</td>
<td>30.34</td>
<td>27.42</td>
<td>39.12</td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>112x112x16</td>
<td>31.24</td>
<td>31.79</td>
<td>44.60</td>
<td></td>
</tr>
<tr>
<td>DRL</td>
<td>112x112x16</td>
<td>33.85</td>
<td>33.24</td>
<td>47.58</td>
<td></td>
</tr>
<tr>
<td>Scratch</td>
<td>3D ResNet-50</td>
<td>31.79</td>
<td>27.90</td>
<td>43.01</td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>112x112x16</td>
<td>33.85</td>
<td>33.24</td>
<td>47.58</td>
<td></td>
</tr>
<tr>
<td>DRL</td>
<td>112x112x16</td>
<td>31.90</td>
<td>32.44</td>
<td>47.00</td>
<td></td>
</tr>
<tr>
<td>Scratch</td>
<td>TSM ResNet-18</td>
<td>30.96</td>
<td>28.21</td>
<td>39.07</td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>224x224x8</td>
<td>31.28</td>
<td>32.29</td>
<td>38.71</td>
<td></td>
</tr>
<tr>
<td>DRL</td>
<td>224x224x8</td>
<td>32.28</td>
<td>31.85</td>
<td>47.04</td>
<td></td>
</tr>
</tbody>
</table>

Table 5. Domain generalization accuracy of Kinetics-pretrained 3D ResNet-18 models, evaluated on 10 classes of Mimetics.

<table>
<thead>
<tr>
<th>Method</th>
<th>Baseline</th>
<th>Learned-Mixin</th>
<th>RUBi</th>
<th>ReBias</th>
<th>DRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy@1</td>
<td>18.9</td>
<td>11.4</td>
<td>13.4</td>
<td>22.4</td>
<td>26.4</td>
</tr>
</tbody>
</table>

**Few-shot recognition.** To evaluate the transferability of the representation as is, we consider the task of cross-domain few-shot classification. That is, a network pretrained on Kinetics (base) classes is used in another dataset (novel classes) to extract features that are fed to a few-shot action recognizer. The target datasets are Something-Something V2 [34] and Jester [58]. This setting presents a greater challenge than most prior work on few-shot video classification (e.g. [12, 92]), as base and novel classes are sampled from different datasets, creating a domain shift which has been found to degrade few-shot learning performance [15]. Few-shot classification is implemented using the Baseline method by Chen et al. [15], which optimizes a linear classifier with the support data in each episode and evaluates it on the query videos. Both 1-shot and 5-shot scenarios are tested, and 5-way accuracy is reported. Table 4 shows DRL outperforms the baseline for all combinations of model and target dataset. It also achieves large gains for most networks on the two target datasets, especially for 5-shot learning. For example, using the TSM ResNet-18 architecture we observed an improvement of 8% on 5-shot action recognition on the Jester dataset. In this setting, three DRL models achieve a rate above 47%, in comparison to a single baseline model above 40%. For Sth-Sth the gains were smaller, but between 2-3% for most models. These results support the conclusion that, despite the large size of Kinetics, models trained on this dataset are somewhat overfitted to spatial cues that do not generalize to other recognition datasets. Reducing this spatial bias, as is done by DRL, increases model robustness and transfer performance.
Domain generalization. We finally evaluate the video action classifiers trained on Kinetics, without fine-tuning, on the Mimetics [87] test set, which contains videos of mimed actions corresponding to 50 Kinetics classes. This leads to a test domain that shares similar action dynamics with the training data, but without co-occurring objects and scenes, reducing the advantages of spatially biased models. We follow the setup of [2] which uses 10 classes of Mimetics for evaluation. As shown in table 5, 3D ResNet-18 models trained with DRL outperforms cross-entropy pretraining baseline and prior debiasing methods by significant margin.

Qualitative examples. A closer look at model predictions in figure 3 reveal that baseline models tend to fail on actions that take place in an unfamiliar environment (e.g. “brushing hair” outdoors, or “surfing” indoors). DRL is capable of correcting these predictions by reducing the reliance on contextual cues (scene, objects etc.) and guiding the network to focus on temporal dynamics of the action itself.

5.4. Ablation studies
Dynamic loss. Figure 4 shows the influence of dynamic loss weight $\lambda$ on the transfer accuracy of learned models, measured by linear probing. We compare three DRL training methods, using either direct min-max optimization (Distill, alg. 2) or min-max approximation by removing temporal information from input clips (Freeze and Shuffle, alg. 3). As the figure depicts, direct min-max optimization suffers from difficult optimization landscape, failing to converge for DRL weight $\lambda > 0.5$, and unable to improve representation quality for smaller values of $\lambda$. Shuffling approximation is effective at improving linear classification of learned representations for small $\lambda$, but the trend reverses as regularization weight increases, eventually leading to poorer results for $\lambda \geq 0.5$. This is likely due to the fact that frame shuffling introduces artifacts of high temporal frequency, which the convolutional networks cannot model. In contrast, min-max approximation by freezing input clips consistently benefits from greater DRL weight $\lambda$, gaining up to 5% in linear classification accuracy on HMDB, 3% on UCF and 1.5% on Diving. These results corroborate the finding from table 2 that DRL with frozen clips also lead to the greatest improvement in dynamic score of trained models.

Input modalities. While the main results have demonstrated that DRL reduces spatial bias and improves transferability of video CNNs that operate on raw RGB inputs, we further experiment with networks that utilize additional modalities to capture motion information explicitly. Table 6 compares the dynamic score and transfer performance of a vanilla two-stream network [70], under different pre-training strategies. It can be observed that two-stream networks suffer from the same type of bias in RGB-based models. This shows that although optical flow streams can be integrated to the networks to enhance motion capturing, they would not explicitly remove spatial bias from the appearance stream. DRL improves dynamic score of the network as well as transfer accuracy on HMDB and Diving, while achieving comparable performances to baseline on UCF.

6. Discussion and Conclusion
In this work we introduced dynamic score, a new measure of temporal dynamics modeling in video convolutional networks. Various methods to evaluate the dynamic score were discussed, either through knowledge distillation to a 2D spatial convnet or by preprocessing input clips to redact temporal information. We also proposed dynamic representation learning (DRL) to improve the dynamic score of networks, using adversarial perturbation or min-max optimization. The so trained video classifiers and their learned representations are empirically evaluated on a set of downstream tasks (linear classification, fine-tuning, few-shot recognition, domain generalization), with results demonstrating a clear benefit of enhanced temporal representation learning.

Acknowledgements. This work was funded in part by NSF awards IIS-1924937, IIS-2041009, a gift from Amazon, and a gift from Qualcomm. We acknowledge the Nautilus platform for some of the experiments discussed above.


