Leveraging Temporal Context in Low Representational Power Regimes

Camilo L. Fosco camilolu@mit.edu
SouYoung Jin souyoung@mit.edu
Emilie Josephs ejosephs@mit.edu
Aude Oliva oliva@mit.edu

MIT CSAIL

Abstract

Computer vision models are excellent at identifying and exploiting regularities in the world. However, it is computationally costly to learn these regularities from scratch. This presents a challenge for low-parameter models, like those running on edge devices (e.g. smartphones). Can the performance of models with low representational power be improved by supplementing training with additional information about these statistical regularities? We explore this in the domains of action recognition and action anticipation, leveraging the fact that actions are typically embedded in stereotypical sequences. We introduce the Event Transition Matrix (ETM), computed from action labels in an untrimmed video dataset, which captures the temporal context of a given action, operationalized as the likelihood that it was preceded or followed by each other action in the set. We show that including information from the ETM during training improves action recognition and anticipation performance on various egocentric video datasets. Through ablation and control studies, we show that the coherent sequence of information captured by our ETM is key to this effect, and we find that the benefit of this explicit representation of temporal context is most pronounced for smaller models. Code, matrices and models are available in our project page: https://camilofosco.com/etm_website

1. Introduction

A strength of computer vision models is their ability to identify and exploit statistical regularities in the world. Learning these regularities from scratch is computationally costly, which limits the accuracy of low-parameters models. It is critical to boost the performance of small models, since many devices lack the resources to host current state of the art models. One way to make the learning problem more tractable for small models may be to supplement them at training time with an explicit representation of the statistical regularities in the target domain. Here, we test whether video understanding models can be improved by providing them with information about typical event sequences during training.

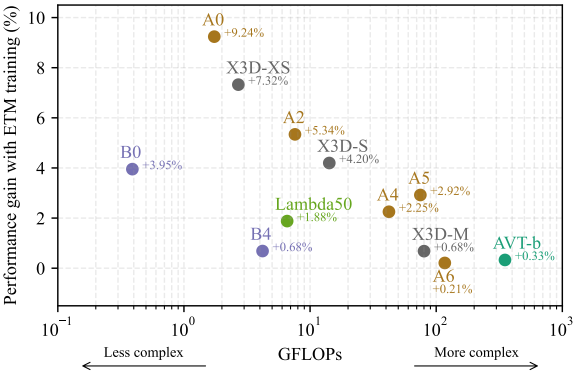
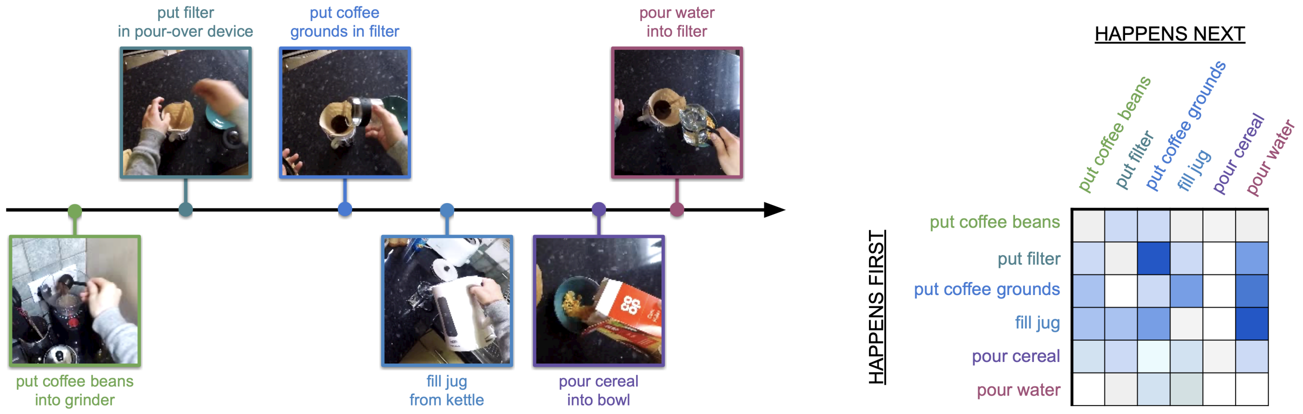


Figure 1. Action recognition performance difference between models trained with and without the proposed ETM approach. We train models with various model architectures, from small (left) to large (right): AVT-b [14], MoViNets [24] family, X3D [11] family, LambdaResNet-50 [4], and EfficientNets [57] family on EPIC-KITCHENS-100 [14]. We show that incorporating the ETM into training improves performance, and the impact is higher on smaller models.

We introduce the Event Transition Matrix (ETM), illustrated in Figure 2, which leverages the fact that events in real-world videos often occur in predictable sequences. Each row and column indexes an event. In the rows, the ETM captures the likelihood that the event was *followed* by each of the other events in the set. In the columns, it captures the likelihood that the event was *preceded* by each other event. To compute a cell's value, we combine information from all previous and subsequent events, weighted by their temporal distance from the queried action. Crucially, this breaks markovian assumptions and allows the matrix to capture long-range relationships. The ETM has two important properties. First, it acts as an explicit representation of the likelihood of event transitions, which provides additional pertinent information that a model can



(a) Low-level descriptions in [10]

(b) Event Transition Matrix

Figure 2. We construct a given Event Transition Matrix (ETM) from action labels drawn from a given dataset of untrimmed videos. (a) Each video depicts a continuous activity, and is paired with human annotations indicating the individual, low-level actions that compose the activity. (b) Using these action labels, we created the ETM by recording the frequency with which a given action label was preceded or followed by each other action label, accumulated across the videos in our set.

leverage without the cost of learning it for itself. Second, it augments representations of a given event with information about what came before and after it, providing additional target for the model to train on.

In the present paper, across multiple datasets and model architectures, we test the effectiveness of incorporating the ETM into training for low-parameter models. We test our approach on action recognition and action anticipation in egocentric video datasets, where actions occur in stereotypical sequences. However, this approach can apply to any kind of sequence in videos. We show that leveraging the ETM improves action recognition relative to baselines, and that this improvement relies on the coherence of the action sequence. We also show that action anticipation is improved with our ETM approach. In both cases, we show that the ETM approach can be incorporated into multiple different model architectures, and that the addition of the ETM has the highest impact on smaller models, as shown in Figure 1. Overall, this work demonstrates a potential path to more efficient models, based on supplementing small models at training time with explicit representations of regularities expected in the data.

2. Related Work

2.1. Efficient AI

State of the art neural networks for video understanding can achieve impressive classification results, but these models often have heavy computational requirements. Producing smaller, more efficient models that can perform comparable to larger ones is an active area of research, for deployment onto edge devices such as phones, wearables, drones or autonomous vehicles. Some approaches achieve efficiency by first training a large network, then using a smaller network to learn the mapping from input to output vectors with little loss on performance (i.e. knowledge distillation [3, 15, 20, 23, 27, 32]), or by identifying and removing non-essential parameters (i.e. parameter pruning [18,19,26,29]). Others rely on adaptive policies to shorten or simplify the inference process, reducing the effective size of the architecture [5, 6, 33, 45] or input [45, 46]. Here, we propose an alternative approach, which involves intervening at training time to supplement low-parameter architectures with external, explicit representations of statistical regularities in the target domain.

2.2. Statistical Regularities and Bayesian Learning.

The environment is full of regularities. Objects classes have co-occurring attributes [34, 50], environments classes have distinctive combinations of objects, surfaces and textures [17, 51, 58, 60], and episodes are composed of predictable sequences of events [25, 63]. Having the ability to detect and leverage these regularities is a key feature of intelligence. Cognitive research in humans has established that representations of contextual regularities underlie most aspects of intelligence, spanning learning, generalization, prediction, compression, memory, language and more [7, 12, 54].

Machine learning relies on identifying and making use of regularities. Typically, this happens in an emergent manner from the training regime. However, some approaches also rely on explicit representations of such regularities. For example, in natural language processing, word embeddings [37, 38, 47, 48, 59] acting as explicit representations of word transition probabilities have provided the basis for many downstream tasks.

2.3. Action Recognition and Anticipation

Action understanding is an enduring problem in video processing [8,41,43,49,64]. This can involve action recognition, which requires retrieving a label for a depicted action, and action anticipation, which requires retrieving a label for a future action based on a depicted action. Videos contain complex semantic relationships between objects, people and places, which evolve dynamically over time. Thus, action understanding models often incorporate elements of relational modeling, for example modeling the spatial relations among people or between people and objects [13, 56], between people and environment zones [44], or between the body parts of the person performing the actions [61]. More recently, multi-modal representation learning approaches have been introduced to learn the broader sensory context of actions [30, 36, 42].

2.4. Action Recognition and Anticipation with Temporal Context

Another kind of context that can be modeled in action understanding is the temporal relationships between the events that make up ongoing actions. Actions take place in sequences: you must first boil water and grind coffee beans, then pour water over the beans in order to successfully make coffee. Some models capitalize on such temporal regularities by modeling the dependence among individual video frames, either with RNNs [2, 56], relational reasoning networks [65] or more recently with attention-based methods and transformers [14,52]. Other approaches model these sequences at a higher level, by learning the dependence among events (rather than individual frames), by using action co-occurrence matrices [22, 40], or by estimating the transitional probability between adjacent [35, 55] or more distant actions [21, 53].

Building on this work, we show how leveraging explicit representations of typical action sequences in low-parameter models can improve their performance. Our motivation is that there is regularity in the temporal context of an action in both the past and the future, and leveraging this bi-directional regularity could bootstrap the model's learning. While some previous work [1] included a representation of the past in an action anticipation framework, our approach is distinct from it in two ways: (1) our representation of temporal context includes past, present and future, and extends over a larger time period, and (2) we pre-compute this representation, then use it to supplement learning.

3. Proposed Approach

In this section, we introduce the Event Transition Matrix (ETM), which is an efficient approach to constructing a knowledge base about the temporal relation between events in video scenes (Section 3.1). We then propose a novel approach to pre-training with this matrix, leveraging these relations to help recognize current events or predict past or future events in the input video snippet (Section 3.2).

3.1. Event Transition Matrix (ETM)

Egocentric video datasets, such as EPIC-KITCHENS [9, 10] or EGO4D [16], contain long form videos that exhibit rich sequences of small actions or events. These sequences are densely annotated with labels for each event and timestamps indicating start and end points of each action. Suppose there are N event categories, $S = \{s_1, s_2, ..., s_N\}$. Each video is annotated with a sequence of event instances, which can be denoted as $\{y_1, y_2, ..., y_{t-1}, y_t, ...\}$. Using the timestamps, we can define a video snippet x_k that contains the specific event, which is given the instance label y_k, based on its category. Event descriptions, i.e. x_k and y_k, refer to the current ongoing action event, but do not convey the broader temporal context of that action, including what came before and after it. As in the example in Figure 2, when making coffee, "add coffee beans" happens earlier than "brew coffee", and these two events are temporally dependent on each other. To leverage the temporal context surrounding events, we compute the frequencies with which each event comes before/after each other event. More formally, we define a square matrix, $\mathbf{M}_{N,N}$, where each row and column corresponds to an event in S and $\mathbf{M}(i, j)$ corresponds to an estimated probability that an event s_i happens before s_j, and vice versa. This matrix is built with training set observations.

A naive approach for building the matrix is to update $\mathbf{M}(y_l, y_m)$ for all possible pairs of events where event l happens earlier than event m. However, this approach does not incorporate the temporal distance between two events, since (y_l, y_m) and (y_l, y_n) contribute equally to constructing the matrix even when l is much farther away from m. In reality, actions separated by more time are less likely to be part of the same overall activity, and are therefore less likely to consistently co-vary. Therefore, we update the matrix with a decay function $\delta(\cdot)$, as

$$\mathbf{M}(y_l, y_m) \mathrel{+}= \delta(m - l). \tag{1}$$

We normalize the matrix $\mathbf{M}$ by the sum of each row and define a new matrix $\mathbf{M}^R$ as

$$\mathbf{M}^R(i, j) = \frac{\mathbf{M}(i, j)}{\sum_{k=0}^{N} \mathbf{M}(i, k)}. \tag{2}$$

Each row in this matrix, $\mathbf{M}^R(i, :)$, can be seen as an empirical approximation of the distribution of possible events that can be transitioned to following event s_i. Similarly, we can also define a matrix $\mathbf{M}^C$ by column-wise normalization.

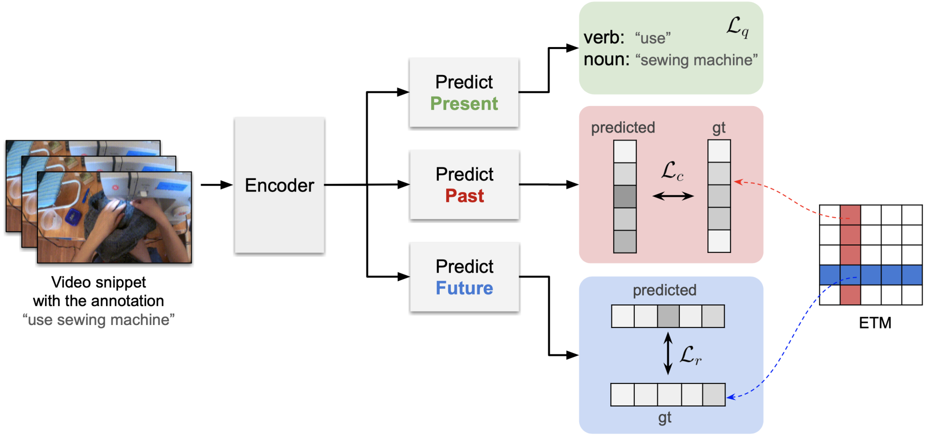


Figure 3. Overview of the proposed approach for pre-training with ETM. The model takes a video snippet and encodes the input to predict the **present**, **past**, and **future** events. In particular, we use the column and the row of the ETM which correspond to the given event. These represent the distributions over past actions, and future actions respectively. More details are written in Section 3.2.

3.2. Encoder Training with the Event Transition Matrix

We propose a novel training protocol to enhance an arbitrary encoder by leveraging the ETM. A typical event recognition model is trained to take a video snippet, x_t, transform it into a feature vector with an encoder $f(\cdot)$, and recognize an event in the input video by predicting a label from that vector. More formally, the model is trained to increase $p(y_t|g_q(f(x_t)))$, where g_q represents a module that maps features into a probability distribution over labels. In our framework, we call g_q the **present** module and train it with a typical cross-entropy loss function, $\mathcal{L}_q$.

Our proposed method adds two additional modules to predict past and future event probabilities, as shown in Figure 3. A **past** module $g_c(\cdot)$ receives $f(x_t)$ and predicts a distribution of events occurring before the current action (which corresponds to a column of the column-normalized ETM, $\mathbf{M}^C$). This module attempts to minimize the following loss:

$$\mathcal{L}_c = \sum_{k=0}^{N} (C(k) - \mathbf{M}^C(y_t, k))^2, \tag{3}$$

where $C(k) = g_c(f(x_t))$. Similarly, a **future** module $g_r(\cdot)$ predicts a distribution of future events, and tries to minimize the distance between the prediction and a row of the row-normalized ETM, $\mathbf{M}^R(:, y_t)$, through the loss:

$$\mathcal{L}_r = \sum_{k=0}^{N} (R(k) - \mathbf{M}^R(k, y_t))^2, \tag{4}$$

where $R(k) = g_r(f(x_t))$. Altogether, our framework augments an arbitrary encoder with these past and future modules and minimizes the following objective during training:

Dataset	Segm.	Actions	τ_a (s)	Metrics (AR, AA)
EK100 [10]	90k	3807	1.0	top-1, rec@5
EGO4D [16]	39.2k	3542	1.0	top-1, rec@5
EGTEA [28]	10.3k	106	0.5	top-1, top-1

Table 1. Properties of the datasets used in our experiments. We showcase our three main datasets: EPIC-KITCHENS-100 (EK100), EGO4D LTA (EGO4D) and EGTEA Gaze+ (EGTEA). τ_a corresponds to the time between input clips and target segments in action anticipation, and follows prior work [14]. The Metrics column shows the main performance metrics used in our action recognition and action anticipation experiments, respectively.

$$\mathcal{L} = \omega_q \mathcal{L}_q + \omega_c \mathcal{L}_c + \omega_r \mathcal{L}_r, \tag{5}$$

This training setup attempts to push the encoder towards outputting richer event representations that capture the regularities of the event's typical temporal context. Although simplistic, the framework is effective: we observe that instantiating g_c and g_r with simple multi-layer perceptrons is enough to achieve strong results on low-complexity models.

4. Experimental Results

In this section, we demonstrate the effectiveness of training with the Event Transition Matrix (ETM) on action recognition (Section 4.3) and action anticipation (Section 4.4). We describe the datasets used in Section 4.1 and give implementation details in Section 4.2.

4.1. Datasets

EPIC-KITCHENS-100. [10] The EPIC-KITCHENS-100 dataset (EK100) contains 700 unscripted videos depicting cooking actions, totalling 100 hours. It presents verb and noun annotations over 90k segments of varying length. The dataset depicts 97 unique verbs, 300 unique nouns that combine to yield 3807 actions. We work with the provided training and validation sets, containing 67.2k and 9.7k segments respectively. Our ETM is constructed exclusively with training set data.

EGO4D. [16] This newer egocentric dataset contains 3670 hours of video from 71 different participants. We use the long term anticipation annotations provided by the authors [16], which reduces the set to a training split containing 493 unique videos with 23610 segments, and a validation set with 380 videos cut into 15587 segments. These annotations present 115 unique verbs, 477 unique nouns and 3542 unique actions. All segments are 240-frame long. To homogenize our setup and avoid generating ETMs with a biased diagonal, we merge contiguous segments together if they depict the same action. This yields 18896 segments for training and 12676 segments for validation, and we use this training set for ETM construction. We refer to this set as EGO4D LTA.

EGTEA Gaze+ [28] is another popular egocentric dataset with 10k segments annotated with 19 verbs, 51 nouns and 106 unique actions. We report performance for split 1 (provided by the authors), containing 8299 training segments and 2022 validation segments. Following previous work [14, 39], both action recognition and anticipation performance are measured with top-1 accuracy. The properties of each dataset are summarized in Table 1.

4.2. Implementation Details

ETM construction. We construct our ETM offline by observing all videos of our training set and adding a contribution to $M(y_l, y_m)$ modulated by a decay function $\delta(\upsilon)$, where υ is the temporal distance between the events, as in Equation 1. Importantly, to minimize sparsity, we only consider events that appear at least 4 times in the training set.

Encoder training. With a given ETM, we train an encoder to generate embeddings that can be used to both recognize actions and predict ETM vectors from snippets of video. The encoder is tasked with producing an embedding that is fed to three modules: an action classification head (present module), a past vector regressor and a future vector regressor, as seen in Figure 3. We instantiate the past, present and future modules with simple fully-connected layers. We train with the loss in Eq. 5 and set $\omega_q = \omega_c = \omega_r = \frac{1}{3}$. As the ETM is built with actions that appear at least four times in the training data, certain rare actions do not have target vectors. In those cases, the loss associated to regressing the past and future vectors is set to zero, and no gradients are back-propagated.

We train our models with the Ranger21 optimizer [62] with a learning rate of 0.01 and cosine annealing with a 20 epochs half-cycle. We use a batch-size of 32, weight decay of 0.0001 and dropout with $p = 0.5$ where applicable.

Training setup for Action Anticipation. We follow the framework used in [14], where an encoder generates features from an input clip c before feeding them to a decoder attempting to predict the future of c, which is the label y_s of an action segment s happening τ_a seconds after c ends. In other words, for each action segment starting at time τ_s, the decoder attempts to predict its action label y_s using a clip c that ends τ_a seconds before s. Crucially, we analyze how using an encoder trained with ETM supervision affects the performance of the system.

We analyze two different setups: one were we freeze our pretrained encoder and only train the decoder, and another where we allow our encoder to be finetuned alongside the decoder. Following the AVT framework [14], we replace AVT-b (AVT's transformer-based encoder) with several variants of our own encoders. We keep AVT-h (AVT's transformer-based head) as a decoder and finetune it with our codes as input.

We train until validation performance plateaus, and we operate with a batch size of 32, a learning rate of 10^{-4} with cosine annealed decay, and the Ranger21 optimizer. At test time, we employ 3-crop testing following [14], where we compute three 224px spatial crops from 248px input frames, and average the predictions of each segment.

4.3. Action Recognition Experiments

We first ask whether models trained with ETM supervision showed improved action recognition performance. To do that, we train a baseline MoViNet A0 on EK100. Following [10], our *present* head has a two-way output: one to predict verbs and one to predict nouns, with an average verb/noun loss. Accordingly, in Table 2, the Present tab shows the top-1 accuracies in verb and noun classification as well as Action accuracy, which corresponds to (verb, noun) pairs. We compare this to a MoViNet A0 trained on the same data but augmented with our ETM framework as described in Section 3. For this model, we also show the mean absolute error (MAE) on the *past* and the *future* predictions. Here, lower scores indicate better performance.

To demonstrate that the performance gain is not merely due to the extra parameters gained from training with an external matrix, we compare against three more baselines, with versions of the matrix that disrupt the action sequence information to different degrees. In the *Full shuffle* baseline, cells of the matrix are randomly shuffled. This scrambles any sequence information in the matrix, while preserving the distribution of values across it. In the *Columns shuffle*, columns are kept intact, but their position in the matrix is randomly shuffled. This preserves individual distributions over past actions, but randomizes their correspondence with columns labels and scrambles the rows (similarly, the *Row shuffle* kept rows intact but scrambled their position in the

Model	**Present**			MAE on	MAE on
	Verb ↑	Noun ↑	Action ↑	**Past** ↓	**Future** ↓
Baseline	64.8	47.4	36.8	-	-
Full shuffle	64.1	47.2	36.3	4.117	4.012
Columns/rows shuffle	64.7	47.6	36.7	3.254	3.101
Co-occurrence	65.3	49.0	37.9	1.211	1.115
Only past vector	65.7	49.3	38.2	0.901	-
Only future vector	65.5	49.8	38.3	-	0.898
ETM (Ours)	**67.9**	**51.2**	**40.2**	**0.882**	**0.859**

Table 2. Action recognition results on various baseline models. We train the models on the EPIC-KITCHENS-100 dataset [14] with the MoViNet A0 [24]. For **Present** prediction, we show the verb and noun classification results and retrieval scores of (verb, noun). For **Past** and **Future** prediction, we show the mean absolute error (MAE) on the past and the future predicted vectors.

Dataset	Model	**Present**		
		Verb	Noun	Action
EK100 [14]	Baseline	64.8	47.4	36.8
	ETM(Ours)	**67.9**	**51.2**	**40.2**
EGO4D	Baseline	32.3	23.5	21.1
LTA [16]	ETM(Ours)	**32.9**	**24.2**	**22.0**
EGTEA	Baseline	81.2	71.7	60.4
Gaze+ [28]	ETM(Ours)	**83.4**	**72.9**	**62.5**

Table 3. Action recognition results on various datasets using MoViNet A0 [24] with and without ETM supervision. As can be seen, augmenting the model with our framework improves performance on all datasets.

Model	w/o ETM	w/ ETM
MoviNet A0 [24]	36.8	**40.2**
MoviNet A2 [24]	41.2	**43.4**
X3D-XS [11]	35.5	**38.1**
X3D-S [11]	40.5	**42.2**
ConvNeXt-S 224 [31]	20.1	**32.4**
LambdaResNet-50 [4]	26.6	**27.1**
EfficientNet-B0 [57]	25.3	**26.3**
EfficientNet-B4 [11]	29.2	**29.4**
AVT-b [14]	30.4	**30.7**

Table 4. Additional action recognition results on EPIC-KITCHENS-100 for models trained with and without ETMs. We show top-1 action classification results for several different architectures.

matrix). Finally, we also construct a matrix which does not consider the order of the events, instead using a symmetric matrix of *co-occurrence* for training the action recognition model. In Table 2, we see that our proposed ETM surpasses all other baselines. We show additional results in the supplementary material.

Next, we further conduct ablation studies to probe whether it is necessary to use both the past and future information from the ETM to see these performance benefits. We train two additional models including *only* either the past or the future module. Interestingly, as shown in Table 2, training models with either side of the temporal information still improves performance to some extent, but we observe that training with the full matrix gives us the biggest improvement. These experiments serve as quantitative justifications for our framework's design choices.

To show that the proposed ETM approach is effective on multiple datasets, we also show in Table 3 the performance gains on two other egocentric activity datasets that contain untrimmed videos of actions annotated with verbs and nouns. Specifically, we train models on the EGO4D dataset [16] using the long term action anticipation annotations, as well as the Extended GTEA Gaze+ (EGTEA Gaze+) dataset [28].

The ETM is model-agnostic and can be incorporated into any model architecture. We illustrate this by showing how this technique performs on a breadth of models: In Table 4, we show action recognition results for multiple model architectures with and without ETM training. Our framework benefits low-complexity models the most. Adding ETM supervision to an advanced encoder like AVT-b [14] produces slight improvements, but small models like EfficientNet-B0 achieve noticeable gains. We show qualitative performance results with correct and incorrect classifications in the supplementary material.

Dataset	Frozen Encoder?	Baseline			ETM (Ours)		
		Verb ↑	Noun ↑	Action ↑	Verb ↑	Noun ↑	Action ↑
EK100	✓	19.9	20.4	7.2	**21.5**	**20.5**	**8.1**
		20.8	21.3	8.0	**22.4**	**22.7**	**9.1**
EGO4D LTA	✓	17.1	16.6	10.3	**18.1**	**17.8**	**11.4**
		18.2	17.5	11.1	**19.9**	**19.1**	**12.9**
EGTEA Gaze+	✓	42.1	37.6	28.9	**43.4**	**38.9**	**31.3**
		43.5	38.5	30.3	**46.5**	**40.7**	**34.1**

Table 5. Action anticipation results using MoViNet A0 [24] as an encoder, trained with and without our ETM protocol. We show results on multiple datasets. Performance is measured with class-mean recall@5 at 1s for EK100 and EGO4D, and top-1 accuracy at 0.5s for EGTEA Gaze+, following previous work [14]

Encoder	w/o ETM	with ETM
MoViNet A0 [24]	8.0	**9.1**
MoViNet A2 [24]	10.2	**10.8**
X3D-XS [11]	6.3	**7.4**
X3D-S [11]	9.4	**9.9**
ConvNeXt-S 224 [31]	4.1	**5.0**
EfficientNet B0 [57]	7.2	**8.0**
EfficientNet B4 [57]	9.4	**10.1**
AVT-b [14]	13.4	**13.5**

Table 6. Additional action anticipation results on EPIC-KITCHENS-100 for encoders trained with and without ETMs. We show class-mean recall@5 results without and with ETM for a variety of encoders. All models are first pretrained on the action recognition task, with and without the ETM protocol.

4.4. Action Anticipation Experiments

We also demonstrate how an encoder pre-trained on our ETM can improve performance in downstream tasks, in this case action anticipation. We follow the anticipation model architecture and the evaluation protocol from [14] and report the accuracies in verb, noun, and action prediction. However, our key difference is that the video encoder was pre-trained on an action recognition task, supplemented with ETM at training time.

Table 5 showcases performance differences between encoders with and without ETM pretraining. Low complexity models, like MoviNet A0 (showcased in Table 5) exhibit improved performance in all three datasets. More comparisons on different model architectures are shown in Table 6 and Figure 4.

4.5. ETM and Model Complexity

To further investigate how model complexity interacts with the proposed ETM training protocol, we train progressively more complex models in the same architecture families and analyze how task performance varies with model complexity. We perform this analysis with the MoViNet [24] and X3D [11] architecture families: MoViNets scale from A0 to A4 with increasing parameter count and GFLOPs (Billion of Floating Point Operations), while X3Ds scale from XS to XXL. We train MoViNets A0 to A4 and X3Ds XS to M for our experiments. We use the same training setup as Section 3.2, where one instance of each model is trained without the ETM and another with it.

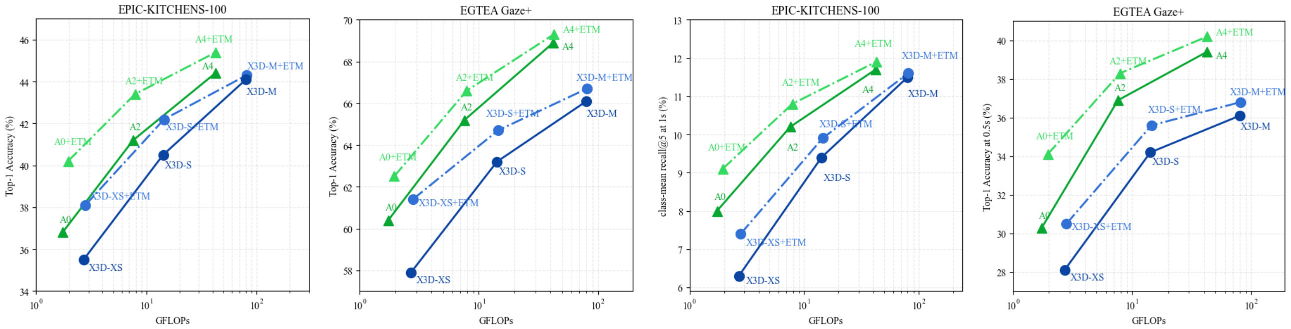
For action recognition, we observe an increase in performance for low complexity models, i.e. A0 to A4, as shown in Figure 4 (a). The gain is larger for lower complexity models. We observe a similar effect for X3D models: the simpler the model is, the larger the improvement appears to be. This effect is visible on both EK100 and EGO4D.

Figure 4 (b) shows that a similar effect occurs with action anticipation: using backbone models trained with our ETM protocol improves performance, especially if the backbone models are small. Once again, we use these models as encoders in the AVT [14] framework, replacing AVT-b (AVT's transformer-based encoder) with these variants and using AVT-h (AVT's transformer-based head) as a head. We only show GFLOPs of the encoders. We observe that on both datasets, encoders with lower representational power benefit more from the information about past and future regularities brought by the ETM.

4.6. ETM Alternatives

Our proposed ETM can take many forms. We analyzed the impact of three of the main properties of the matrix on action recognition performance: its size, the decay function used to build it, and the metric for temporal distance between events.

Matrix Size. Computing the matrix with every unique event (verb-noun combination) appearing in the training set of EK100 yields a $13k \times 13k$ matrix. Since single-instance



(a) Action Recognition

(b) Action Anticipation

Figure 4. Performance for models on EK100 [14] and EGTEA Gaze+ [28] in the same architecture family. We see that for both MoViNets [24] and X3Ds [11], leveraging the ETM during training improves performance, and the effect is stronger with lower complexity models. MoViNet+ETM (— -▲); MoViNet (——▲); X3D+ETM (— -●); X3D (——●).

actions may be noisier, we reduced the dimensionality of the matrix by removing events with less than 4 occurrences in the training set, yielding a subset of 2562 actions and a matrix of size 2562×2562. We compare both alternatives in Table 7.

Decay function alternatives. We compared linear decay, exponential decay and no decay. We show comparison results in Table 7.

Measuring temporal distance. We considered the natural alternative of measuring temporal distance between events A and B in seconds, by looking at the time between the last frame of A and the first frame of B. We observed, however, that the durations of individual actions between A and B could vastly modulate the decay factor applied to $M(y_A, y_B)$ of our matrix, which might be counterproductive if A and B are causally related but simply separated by one long intermediate action. We therefore considered measuring distance through *index difference* in the sequence of actions for that video, which allows us to abstract away the durations of individual actions.

We compared these alternatives based on their ability to boost performance on action recognition. We trained a ConvNeXt-S (224) [31] encoder with supervision from different ETM alternatives and evaluated action recognition performance (Table 7). Our results show that the choice of decay is generally not impactful unless no decay is used. We find our best performance on a matrix of reduced size, with index as a temporal distance metric, and with exponential decay.

Size	Decay	Temp. Metric	Present (top-1 accuracy)		
			Verb	Noun	Action
13k	Linear	Time	55.1	46.2	28.8
13k	Exponential	Time	55.6	47.7	29.1
2.5k	Exponential	Time	58.6	48.8	31.3
2.5k	No decay	-	58.1	48.0	30.5
2.5k	Linear	Index	60.1	49.3	31.9
2.5k	Exponential	Index	**60.3**	**50.3**	**32.4**

Table 7. Difference in action recognition performance between ETM alternatives. We train ConvNeXts on EPIC-KITCHENS-100 with different verions of our ETM to evaluate the ability of ETM to improve action recognition performance. We evaluate top-1 accuracy on action recognition over the validation set of EK100.

5. Conclusions

We test a novel training regime for video understanding, in which model performance can be increased by supervising models with external representations of temporal regularities. We show that using the ETM as a training target allows arbitrary models to learn about the bi-directional temporal context of the action (i.e. the past and future), which improves action recognition and action anticipation performance. Finally, we tested the circumstances under which the ETM provides the largest benefit, and find significant performance boosts for low-complexity models. Amid ongoing efforts to increase the efficiency of computer vision models, we suggest that research which explores efficient architectures [11, 24] could be complemented by research that explores ways to leverage pre-learned representations of environmental regularities.

The biggest benefit of this data structure is in its flexibility and simplicity: it is model-agnostic, computationally inexpensive, and easy to incorporate into the learning regime of any action recognition or anticipation architecture.

Acknowledgements. This work was partially funded by a grant from Meta. We thank Karl Ridgeway and the team at Meta Reality Labs Research for their helpful comments.

References

[1] Yazan Abu Farha, Qiuhong Ke, Bernt Schiele, and Juergen Gall. Long-term anticipation of activities with cycle consistency. In *DAGM German Conference on Pattern Recognition*, pages 159–173. Springer, 2020. 3

[2] Yazan Abu Farha, Alexander Richard, and Juergen Gall. When will you do what?-anticipating temporal occurrences of activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5343–5352, 2018. 3

[3] Hande Alemdar, Vincent Leroy, Adrien Prost-Boucle, and Frédéric Pétrot. Ternary neural networks for resource-efficient ai applications. In *2017 international joint conference on neural networks (IJCNN)*, pages 2547–2554. IEEE, 2017. 2

[4] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3286–3295, 2019. 1, 6

[5] Emmanuel Bengio, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup. Conditional computation in neural networks for faster models. *arXiv preprint arXiv:1511.06297*, 2015. 2

[6] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 2

[7] Michael R Brent. Speech segmentation and word discovery: A computational perspective. *Trends in Cognitive Sciences*, 3(8):294–301, 1999. 2

[8] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4724–4733, 2017. 3

[9] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018. 3

[10] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 2021. 2, 3, 4, 5

[11] 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. 1, 6, 7, 8

[12] Vanessa E Ghosh and Asaf Gilboa. What is a memory schema? a historical perspective on current neuroscience literature. *Neuropsychologia*, 53:104–114, 2014. 2

[13] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2019. 3

[14] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13505–13515, 2021. 1, 3, 4, 5, 6, 7, 8

[15] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021. 2

[16] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the World in 3,000 Hours of Egocentric Video. *CoRR*, abs/2110.07058, 2021. 3, 4, 5, 6

[17] Michelle R Greene and Aude Oliva. Recognition of natural scenes from global properties: Seeing the forest without representing the trees. *Cognitive psychology*, 58(2):137–176, 2009. 2

[18] Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural networks. *CoRR*, abs/1506.02626, 2015. 2

[19] Babak Hassibi and David Stork. Second order derivatives for network pruning: Optimal brain surgeon. *Advances in neural information processing systems*, 5, 1992. 2

[20] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network (2015). *arXiv preprint arXiv:1503.02531*, 2, 2015. 2

[21] Qiuhong Ke, Mario Fritz, and Bernt Schiele. Time-conditioned action anticipation in one shot. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9925–9934, 2019. 3

[22] Dong-Jin Kim, Xiao Sun, Jinsoo Choi, Stephen Lin, and In So Kweon. Detecting human-object interactions with action co-occurrence priors. In *European Conference on Computer Vision*, pages 718–736. Springer, 2020. 3

[23] Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*, 2016. 2

[24] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. Movinets: Mobile video networks for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16020–16030, 2021. 1, 6, 7, 8

[25] Christopher A Kurby and Jeffrey M Zacks. Segmentation in the perception and memory of events. *Trends in cognitive sciences*, 12(2):72–79, 2008. 2

[26] Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989. 2

[27] Jinyu Li, Rui Zhao, Jui-Ting Huang, and Yifan Gong. Learning small-size dnn with output-distribution-based criteria. In *Fifteenth annual conference of the international speech communication association*, 2014. 2

[28] Yin Li, Miao Liu, and James M. Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *ECCV*, 2018. 4, 5, 6, 8

[29] Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461:370–403, 2021. 2

[30] Alexander H Liu, SouYoung Jin, Cheng-I Jeff Lai, Andrew Rouditchenko, Aude Oliva, and James Glass. Cross-modal discrete representation learning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, May 2022. 3

[31] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022. 6, 7, 8

[32] Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher–student curriculum learning. *IEEE transactions on neural networks and learning systems*, 31(9):3732–3740, 2019. 2

[33] Yue Meng, Chung-Ching Lin, Rameswar Panda, Prasanna Sattigeri, Leonid Karlinsky, Aude Oliva, Kate Saenko, and Rogerio Feris. Ar-net: Adaptive frame resolution for efficient action recognition. In *European Conference on Computer Vision*, pages 86–104. Springer, 2020. 2

[34] Carolyn B Mervis and Eleanor Rosch. Categorization of natural objects. *Annual review of psychology*, 32(1):89–115, 1981. 2

[35] Antoine Miech, Ivan Laptev, Josef Sivic, Heng Wang, Lorenzo Torresani, and Du Tran. Leveraging the present to anticipate the future in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 3

[36] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, pages 2630–2640, 2019. 3

[37] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 2

[38] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013. 2

[39] Kyle Min and Jason J Corso. Integrating human gaze into attention for egocentric activity recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1069–1078, 2021. 5

[40] Shayan Modiri Assari, Amir Roshan Zamir, and Mubarak Shah. Video classification using semantic concept co-occurrences. In *CVPR*, pages 2529–2536, 2014. 3

[41] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, and Aude Oliva. Moments in time dataset: one million videos for event understanding. *IEEE TPAMI*, 42(2):502–508, 2019. 3

[42] Mathew Monfort, SouYoung Jin, Alexander Liu, David Harwath, Rogerio Feris, James Glass, and Aude Oliva. Spoken moments: Learning joint audio-visual representations from video descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14871–14881, June 2021. 3

[43] Mathew Monfort, Bowen Pan, Kandan Ramakrishnan, Alex Andonian, Barry A Mcnamara, Alex Lascelles, Quanfu Fan, Dan Gutfreund, Rogerio Feris, and Aude Oliva. Multi-moments in time: Learning and interpreting models for multi-action video understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. 3

[44] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. Ego-topo: Environment affordances from egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 163–172, 2020. 3

[45] Bowen Pan, Rameswar Panda, Camilo Fosco, Chung-Ching Lin, Alex Andonian, Yue Meng, Kate Saenko, Aude Oliva, and Rogerio Feris. Va-red2: Video adaptive redundancy reduction. *arXiv preprint arXiv:2102.07887*, 2021. 2

[46] Bowen Pan, Rameswar Panda, Yifan Jiang, Zhangyang Wang, Rogerio Feris, and Aude Oliva. Ia-red2: Interpretability-aware redundancy reduction for vision transformers. *Advances in Neural Information Processing Systems*, 34:24898–24911, 2021. 2

[47] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 2

[48] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations, 2018. 2

[49] Ivan Rodin, Antonino Furnari, Dimitrios Mavroeidis, and Giovanni Maria Farinella. Untrimmed action anticipation. *arXiv preprint arXiv:2202.04132*, 2022. 3

[50] E Rosch, C B Mervis, W. D. Gray, D. M. Johnson, and P Boyes-Braem. Basic objects in natural categories. *Cognitive psychology*, 8(3):382–439, 1976. 2

[51] Zahra Sadeghi, James L McClelland, and Paul Hoffman. You shall know an object by the company it keeps: An investi-

gation of semantic representations derived from object co-occurrence in visual scenes. *Neuropsychologia*, 76:52–61, 2015. 2

[52] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long-range video understanding. In *European Conference on Computer Vision*, pages 154–171. Springer, 2020. 3

[53] Yang Shen, Bingbing Ni, Zefan Li, and Ning Zhuang. Egocentric activity prediction via event modulated attention. In *Proceedings of the European conference on computer vision (ECCV)*, pages 197–212, 2018. 3

[54] Brynn E Sherman, Kathryn N Graves, and Nicholas B Turk-Browne. The prevalence and importance of statistical learning in human cognition and behavior. *Current opinion in behavioral sciences*, 32:15–20, 2020. 2

[55] Bilge Soran, Ali Farhadi, and Linda Shapiro. Generating notifications for missing actions: Don't forget to turn the lights off! In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4669–4677, 2015. 3

[56] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Rahul Sukthankar, Kevin Murphy, and Cordelia Schmid. Relational action forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 273–283, 2019. 3

[57] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 1, 6, 7

[58] Antonio Torralba and Aude Oliva. Statistics of natural image categories. *Network: computation in neural systems*, 14(3):391, 2003. 2

[59] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*, 2019. 2

[60] Melissa Le-Hoa Võ. The meaning and structure of scenes. *Vision Research*, 181:10–20, 2021. 2

[61] Daniel Weinland and Edmond Boyer. Action recognition using exemplar-based embedding. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE, 2008. 3

[62] Less Wright and Nestor Demeure. Ranger21: a synergistic deep learning optimizer. *arXiv preprint arXiv:2106.13731*, 2021. 5

[63] Jeffrey M Zacks and Barbara Tversky. Event structure in perception and conception. *Psychological bulletin*, 127(1):3, 2001. 2

[64] Haotong Zhang, Fuhai Chen, and Angela Yao. Weakly-supervised dense action anticipation. *arXiv preprint arXiv:2111.07593*, 2021. 3

[65] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. *European Conference on Computer Vision*, 2018. 3