Learning Temporal Dynamics from Cycles in Narrated Video

Dave Epstein
UC Berkeley*

Jiajun Wu
Stanford University

Cordelia Schmid
Google

Chen Sun
Google, Brown University

Abstract

Learning to model how the world changes as time elapses has proven a challenging problem for the computer vision community. We introduce a self-supervised approach to this problem that solves a multi-modal temporal cycle consistency objective jointly in vision and language. This objective requires a model to learn modality-agnostic functions to predict the future and past that undo each other when composed. We hypothesize that a model trained on this objective will discover long-term temporal dynamics in video. We verify this hypothesis by using the resultant visual representations and predictive models as-is to solve a variety of downstream tasks. Our method outperforms state-of-the-art self-supervised video prediction methods on future action anticipation, temporal image ordering, and arrow-of-time classification tasks, without training on target datasets or their labels.

1. Introduction

Prediction is a central problem in computer vision which researchers have been grappling with since the early days of the field [10, 12, 22, 29, 34, 39, 54]. Previous deep learning methods have largely focused on predicting fixed, small offsets into the future. To understand why this formulation is flawed, consider Figure 1. This figure shows a frame **(a)** from a video at time t and three frames at times $> t$. Which of the three should be the output of a model that predicts the future? Option **(d)** is closest to the future that humans are likely to imagine. By predicting frames such as option **(b)**, which occur in the immediate future [15, 16, 47, 53], we limit the scope of temporal transitions that can be learned by models and hurt downstream performance.

Motivated by this example, we identify three central challenges in training a model to predict the future. First, manually annotating videos with temporal relationships between frames is prohibitively expensive, and ground truth may be difficult to define. Therefore, models should be able to learn from large unlabeled datasets of in-the-wild action and discover transitions autonomously, to enable practical applications. Second, modeling the complex long-term transitions in the real world requires learning high-level concepts, more naturally found in abstract latent representations than raw pixels. Finally, the duration elapsed by temporal transitions can vary significantly depending on context, and models must be able to make predictions at varied offsets into the future. To satisfy these desiderata, we introduce a new self-supervised training objective, **M**ulti-**M**odal Temporal **C**ycle **C**onsistency (MMCC), and a model that learns a representation to solve it.

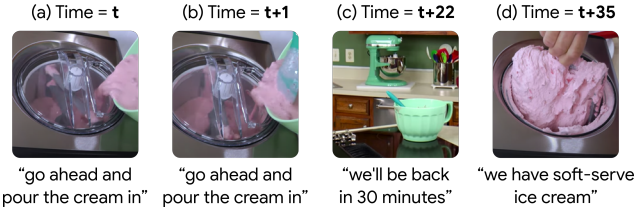

Figure 1. **Predicting the future is challenging.** Given a frame **(a)** at time t, previous work focuses on predicting frames at a fixed offset, such as **(b)**. However, these frames are often either redundant or stochastic, motivating the prediction of non-immediate futures. Predicting such a frame is highly non-trivial, as many are irrelevant, such as **(c)**. Aided by the textual information in narrated video, we can learn long-term temporal dynamics in video, and predict **(d)**. We learn these dynamics by solving a multi-modal temporal cycle consistency problem.

We show the MMCC objective in Figure 2. Starting from a sampled frame in a narrated video, our model learns to attend among all narration text to retrieve a relevant utterance. Combining both modalities, the model learns a function to predict a latent future, attending over the entire video to retrieve a future frame. This frame's corresponding utterance is estimated, and a function to predict a past frame is learned in a similar way. The cycle constraint requires that the final model prediction be equal to the starting frame.

MMCC addresses all three challenges discussed above. In Figure 1, only **(d)** is a viable solution to our cycle formulation. Selecting **(c)** as a future would not allow the model to return to **(a)**, since the two frames have no clear relation-

* Work done as an intern at Google Research.

ship. On the other hand, because the model does not know which modality its input comes from—and therefore must operate equally on vision and language—it is discouraged from selecting lower-level future frames such as **(b)**, which likely do not accompany a predictable change in text.

We show that our model, trained end-to-end from scratch to solve the MMCC objective on the HowTo100M dataset [37], captures long-term dynamics in its predictive model of the future, and can be used without further training to anticipate future actions, order image collections, and identify salient temporal relationships in long videos. It also learns representations of video and text that contain information relevant to modeling temporal dynamics, which we demonstrate to be crucial to the quality of prediction.

Our main contributions are:

- MMCC, a self-supervised multi-modal temporal cycle consistency objective that requires learning visual representations attuned to temporal dynamics, as well as long-term predictive models of the future and past.
- An attention-based model to solve this objective, which uses cross-modal and temporal cues to discover relationships through time in video.
- Since no previous self-supervised benchmarks exist in this area, a suite of qualitative and quantitative tasks to evaluate learned representations and predictive models. Our model outperforms the self-supervised SOTA in video prediction on all tasks.

2. Related Work

Modeling the future. Building predictive models of the future is a long-studied task in the computer vision community. Early work considers generating or warping pixels or optical flow to synthesize immediate futures [3, 11, 32, 41, 42, 46, 54, 55, 56, 57, 64]. More recent work attempts to model uncertainty in pixel-space, often by learning a distribution of futures that can be sampled [8, 18, 21, 27, 29, 48, 51, 52, 62]. These approaches tend to focus on synthetic or very short-term data, since synthesis is challenging in real video. Rather than predicting pixels, another line of work uses supervision to predict future action labels [19, 24, 28, 36, 45]. Sun *et al*. [49] also uses narrated video, but quantizes input video using Kinetics supervision, then learns a transformer-based model of vision-and-language sequences. Instead of using supervision, Vondrick *et al*. [53] predicts *representations* which are trained to capture abstract concepts but are automatically obtained on large collections of data. Recent work extends this, using contrastive learning or other techniques to predict future representations [13, 15, 16, 47, 60]. With very few exceptions [21], this line of work is concerned with predicting

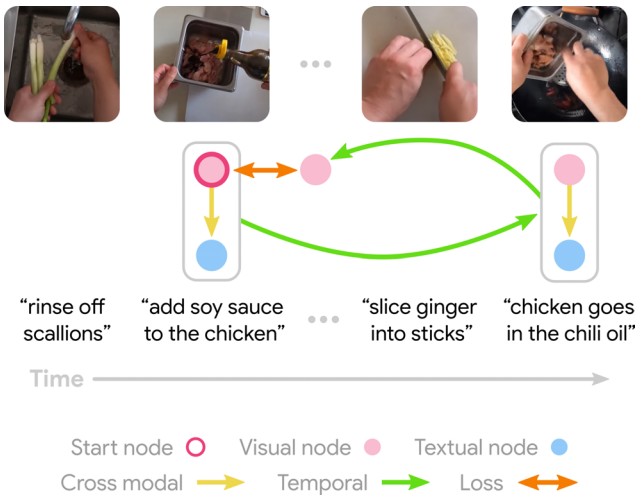

Figure 2. **Learning temporal dynamics with cycles.** Given an image (here, second from left) as a start node, our model finds corresponding text to build a start state. From this, our model predicts a future image and again builds a multi-modal state. Finally, our model predicts a past image from the future state. The discrepancy between this prediction of the past and the start image gives our **cycle-consistency loss**. To solve this problem, we learn the **temporal** and **cross-modal** edges using soft attention.

time $t+1$ given time t. This formulation is highly constraining. Our model can predict arbitrarily far into the future and learns long-term dynamics from unlabeled, narrated video.

Learning from unlabeled narrated video. Self-supervised learning has a long history, even dating back to the early 1990s, where De Sa [6] considered audiovisual data to "derive label[s] from a co-occurring input to another modality". We join an increasingly popular line of work and leverage automatic textual transcripts extracted from narrated videos uploaded online. Combining video and text has been widely explored in the deep learning era, with datasets largely focusing on manual textual annotation of video [2, 5, 61, 65] or on movies which have provided scripts [43, 44]. Other work instead learns from automatic transcripts of narrations in instructional videos [1, 33, 63]. A main benefit of learning from unlabeled video is that it unlocks unprecedented scales of data; Miech *et al*. [37] introduces a dataset of over 100 million video clips and their narration transcripts, which is later used to learn strong models of cross-modal correspondence [35]. We are inspired by their success in training vision-and-language models on large collections of narrated video, and build on their data and approach to learn temporal dynamics.

Learning with self-supervised cycles. Cycle consistency was recently proposed [66] as a natural cue for learning from unlabeled data or when ground truth is unavailable. In Zhu *et al*. [67], cycles are used for unpaired image-to-image translation; Recycle-GAN [4] builds on this in follow-up work that incorporates simple temporal predic-

tion (one timestep into the future) into these cycles. Kulkarni *et al*. [26] uses cycles to learn mappings between canonical 3D surfaces and 2D images. Dwibedi *et al*. [9] uses cycles to enforce that moments from two different videos should be mutual nearest neighbors, aligning action sequences and learning features useful for downstream tasks. Another line of work uses cycles to track objects through time [20, 58], tracking a pixel forward and then backward in time and requiring that the final pixel be the same as the start pixel. We are inspired by all these applications and introduce a new type of temporal cycle, one which not only incorporates multi-modal information into its learning, but also predicts dynamically into the future, instead of at a fixed offset. In particular, we draw inspiration from Jabri *et al*. [20], which casts temporal edges as contrastive comparisons (*i.e*., attention) among candidate nodes.

3. Learning to Cycle through Narrated Video

Our model learns long-term temporal dynamics by cycling through narrated video. We formulate the cycle consistency problem as follows: Given a moment M_i in a start modality M (either video V or text T), retrieve a corresponding moment in the other modality M', then use both modalities to select a future moment in M. From this future moment, find a correspondence in M', then select a past moment in M. For the cycle to be complete, this final moment must be the same as the initial moment M_i. We illustrate the cycle in Figure 2. Solving this problem requires learning forward- and backward-in-time predictive functions that invert each other, as well as image and sentence embeddings that capture inter-modal correspondences and temporally relevant information.

3.1. Cycles as repeated soft attention

Let $V_{t_0:t_1}$ and $T_{t_0:t_1}$ be sequences of video and text, respectively, drawn from some temporal interval $[t_0, t_1]$. These sequences can be discretized into frames $\{V_i\}_{i=1}^{N_V}$ and utterances $\{T_i\}_{i=1}^{N_T}$, where N_V, N_T are the number of instances the sequence is split into. We refer to each instance as a node, which allows viewing the training goal as learning a cyclic path through a graph, as depicted in Figure 2.

In order to differentiate through the cycle generation process, let uv be an edge in the graph shown in Figure 2. We implement edges as soft retrievals of v given u, as shown in Figure 3. This soft retrieval operation can be viewed as an application of the well-known attention mechanism [50].

We start by running all visual and textual nodes through embedding networks Φ_T and Φ_V initialized with random weights. We use the architecture from [35] for embedding text nodes and a ResNet-18 [17] for visual nodes. This operation yields series of embeddings $\{z_{V_i}\}_{i=1}^{N_V}$ and $\{z_{T_i}\}_{i=1}^{N_T}$.

We then compute the cycle edges, where each edge is an instance of soft attention as described above. The attention operation $\mathbf{Attn}(Q, K, V)$ accepts sets of query, key, and value vectors $Q \in \mathbb{R}^{N_Q \times d}, K, V \in \mathbb{R}^{N_K \times d}$ and returns a set of new values $Z \in \mathbb{R}^{N_Q \times d}$, computed (with τ-temperature softmax along the second dimension) as

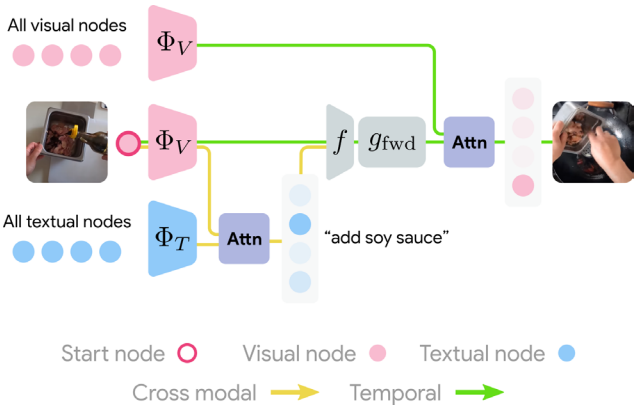

Figure 3. **Structure of a cycle edge:** We learn to embed all visual and textual nodes with Φ_V and Φ_T. We then compute the **cross-modal node** corresponding to the start node with attention across the other modality. Both node representations are passed into an MLP $g_{\text{fwd}} \circ f$, which **predicts the future** using attention across the start modality. The process is then repeated to **go backward in time**, replacing g_{fwd} with g_{back}. Our loss trains the model's final output to close the cycle by predicting the start node.

$$Z = \mathbf{Attn}\left(Q, K, V\right) = \text{Softmax}\left(\frac{QK^T}{\tau}\right)V. \quad (1)$$

To cycle through narrated video, we first select a modality $M \in \{T, V\}$ and a start node $M_\alpha \in M_{t_0:t_1}$ (we describe the process for selecting M_α in Section 3.3). We find the representation of the corresponding node in the other modality M' with a cross-modal attention edge:

$$z_{M'_\alpha} = \mathbf{Attn}\left(\pi_{M_\alpha}, \{\pi_{M'_i}\}_{i=1}^{N_{M'}}, \{z_{M'_i}\}_{i=1}^{N_{M'}}\right). \quad (2a)$$

We learn to project representations z into a shared semantic space, using a modality-specific projector $\Pi_M : \mathbb{R}^d \to \mathbb{R}^d$ that outputs vectors π. For notational convenience, we denote by $\mathbf{Attn}_{n_0:n_1}$ the same attention operation which considers keys (and values) at indices $\{n_0, \ldots, n_1\}$. In this notation we can rewrite the above as:

$$z_{M'_\alpha} = \mathbf{Attn}_{1:N_{M'}}\left(\pi_{M_\alpha}, \{\pi_{M'_i}\}, \{z_{M'_i}\}\right). \quad (2b)$$

The representations from both modalities are concatenated and run through a multi-layer perceptron $f : \mathbb{R}^{2d} \to \mathbb{R}^d$. This operation yields $z_\alpha = f(z_{M_\alpha}, z_{M'_\alpha})$, embedding the joint information back into the shared semantic space. This choice also allows us to train our temporal edges without cross-modal information and accept input from only one

modality with some probability p_{unimodal}, since Φ_V and Φ_T also map to z-space, in $\mathbb{R}^d$.

Our model must now go from this multi-modal state representation z_α to a future state representation z_β. First, we predict an estimated representation of the future in projection (π) space, with an MLP g_{fwd}. We then retrieve the node in modality M corresponding to this future state with a forward-in-time attention edge:

$$z_{M_\beta} = \mathbf{Attn}_{1:N_M}\left(g_{\text{fwd}}(z_\alpha), \{\pi_{M_i}\}, \{z_{M_i}\}\right). \quad (3)$$

It is important to note that attention is order-invariant in K and V, *i.e.* shuffling rows of K and V yields the same output Z, since the individual matrix-row multiplications are agnostic to row index. This means that, importantly, *the model is not given temporal information about input nodes*, which could be used as a shortcut in learning[2]. We then retrieve the corresponding node in M', $z_{M'_\beta}$, with another cross-modal edge, projecting queries and keys into π-space:

$$z_{M'_\beta} = \mathbf{Attn}_{1:N_{M'}}\left(\pi_{M_\beta}, \{\pi_{M'_i}\}, \{z_{M'_i}\}\right). \quad (4)$$

As before, these vectors are combined to yield a future state representation $z_\beta = f(z_{M_\beta}, z_{M'_\beta})$.

This process is repeated to predict backward in time. We compute $g_{\text{back}}(z_\beta)$, where g_{back} shares its first few layers with g_{fwd} to allow learning features useful for dynamics in either direction (see Section 3.5 for more details). To close the cycle, we compute the normalized similarity scores between $g_{\text{back}}(z_\beta)$ and the π-space nodes in M:

$$\mathbf{p} = \mathbf{Attn}_{1:N_M}\left(g_{\text{back}}(z_\beta), \{\pi_{M_i}\}, \{1\}\right). \quad (5)$$

We train our system with the negative log likelihood loss on the score vector cycling back to the location of M_α, which we denote i_α:

$$\mathcal{L}_{\text{cycle}} = -\log \mathbf{p}^{(i_\alpha)}. \quad (6)$$

3.2. Cross-modal correspondence

A key component of our cycle model is the ability to find correspondences between vision and language. Eqs. 2b and 4 crucially rely on this ability in order to incorporate multi-modal information into temporal edges. Recent work has demonstrated remarkable progress in training models on massive datasets for this cross-modal retrieval task: given a moment at time t in one modality of a video - M_t - find the matching moment in the other modality M'.

We build on the approach presented in [35] which uses a contrastive loss to train representations of vision and language, where temporally co-occurring information is considered ground truth (M_t should retrieve M'_t), and other vision-language pairs are used as negatives. To handle the common misalignment intrinsic to real-world video, [35] allows for representations within k nodes of the ground truth node M'_t to be considered as positives. We adopt this approach to learn cross-modal correspondence, training it for finer-grained discrimination among a set of candidate moments drawn from the same video as opposed to randomly across the entire dataset. We denote the loss used to train cross-modal correspondence $\mathcal{L}_{\text{cross-modal}}$. For the full cross-modal formulation, please see Supplementary Material.

[2] *E.g.*, the model could cycle by selecting the node it knows is at $t+1$ or $t-1$.

3.3. Starting the cycle

Our model will be unable to learn semantic transitions between states if the initial input node depicts noisy or unclear data. This is especially probable when training on unconstrained, real-world video datasets. Therefore, instead of randomly sampling start nodes, we sample from a distribution defined by a "concreteness" score s_i. We calculate this score for each node $M_i \in M$ as the highest cross-modal similarity between M_i and some node M'_j in the other modality. Intuitively, this score captures concreteness since frames and utterances that align strongly tend to contain objects or actions which are salient in both modalities:

$$s_i = \max_j \left(\pi_{M_i} \cdot \pi_{M'_j}\right) \quad (7)$$

We run the above scores through a softmax with $\tau = 0.1$, yielding a distribution from which we sample M_α.

3.4. Avoiding collapse

Training on the above formulation of $\mathcal{L}_{\text{cycle}}$ in practice may lead to fast collapse to a simple "looping in place" solution, where temporal edges always point to the current node. We propose two strategies to prevent this collapse:

Constraining candidate nodes. We can limit the range of temporal edges during training by removing nodes from K and V in Eqs. 3 and 5. We rewrite Eq. 3 with $\mathbf{Attn}_{i_\alpha+1:N_M}$, *i.e.*, since we know the index the cycle starts from, we can consider only those nodes after the start point in the forward edge. We similarly rewrite Eq. 5 with $\mathbf{Attn}_{1:\texttt{index}(z_{M_\beta})-1}$, where $\texttt{index}(z_{M_\beta}) = \arg\max_i \left(g_{\text{fwd}}(z_\alpha) \cdot \pi_{M_i}\right)$, *i.e.*, the index of the node with highest similarity to the latent predicted future. This constrains the backward edge to only consider nodes that precede the estimated current index. This can also be seen as resolving the sign ambiguity inherent to the unconstrained formulation which allows the model to go back-then-forward or vice versa. Importantly, we run the model *without* this constraint at test time.

Penalizing visual similarity. Alternatively, we can encourage our model to select visually diverse nodes in its temporal edges:

$$\mathcal{L}_{\text{sim}} = \max\left(\cos(z_{V_\alpha}, z_{V_\beta}) - m, 0\right) + \max\left(\cos(z_{V_\beta}, z_{V_{\alpha,\text{back}}}) - m, 0\right), \quad (8)$$

where $z_{V_{\alpha,\text{back}}}$ is the visual representation given by Eq. 5, replacing the values with $\{z_{M_i}\}$, and $m = 0.5$ is a margin.

In practice, we combine both the above strategies for the strongest results.

3.5. Implementation

We combine $\mathcal{L}_{\text{cycle}}$, $\mathcal{L}_{\text{cross-modal}}$, and $\mathcal{L}_{\text{sim}}$ in our final loss:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{cycle}} + \lambda_2 \mathcal{L}_{\text{cross-modal}} + \lambda_3 \mathcal{L}_{\text{sim}}. \quad (9)$$

We embed images into $\mathbb{R}^d$ ($d = 512$) using a ResNet-18 [17], and embed text using a word embedding matrix followed an MLP and global pooling, as in [35].

We implement all modules (Π_M, f, g_{fwd}, g_{back}) as MLPs, where each layer is followed by ReLU and a LayerNorm except for the final layer, which is followed by l_2 normalization if its output is in π-space. Π_M and f are one-layer MLPs, and $g_{\text{fwd}}, g_{\text{back}}$ are four-layer MLPs, with weights of the first two layers shared. We randomly sample batches of video segments of maximum duration $t_1 - t_0 = 64$sec. The sparsity at which data is sampled affects the time elapsed by input videos in a batch as well as the granularity of visual information provided to the model. Denser data is less likely to miss key moments, but more likely to contain redundant information. We therefore train models on various image sampling frame rates $r \in \{0.25\text{fps}, 0.5\text{fps}, 1\text{fps}\}$.

Because good cross-modal correspondence is necessary to learn strong, semantic cycles, we initialize $\lambda_2 = 1$ and exponentially increase λ_1 from some small value ϵ up to 1, across 30 epochs. We peg $\lambda_3 = 3\lambda_1$ when using the similarity loss. For further details on training and architecture, please see Supplementary Material.

4. Experiments

This sections examines the design choices and learned temporal dynamics of our model. Since most previous benchmarks focus on supervised action anticipation with fixed categories and time offsets [5, 25], we design a suite of qualitative and quantitative experiments to evaluate different approaches.

4.1. Data

We train our model on unconstrained real-world video data. Specifically, we use a subset of the HowTo100M dataset [37], which contains around 1.23 million videos and their automatically extracted audio transcripts. Videos in this dataset are roughly categorized by subject area, and we use only the videos categorized "Recipe", around a quarter of the dataset. We build a train-validation-test split such that of 338,033 total recipe videos, 80% are in train, 15% in validation, and 5% in test. Recipe videos are rich in complex objects, actions, and state transitions, and the subset allows us to train models faster.

For more controlled testing, we use the CrossTask dataset [68], which contains similar videos along with task-specific annotations. Videos are associated with tasks (*e.g.*, "making pancakes"), where each task has a predefined sequence of high-level subtasks with rich long-term temporal inter-dependencies (*e.g.*, ["pour flour into bowl", "crack egg into bowl", ..., "drizzle maple syrup"]). Video segments that depict one of these subtasks are annotated as such.

4.2. Previous work and baselines

Baselines: We evaluate purely cross-modal features (Section 3.2), given by frozen embedding nets Ψ_V, Ψ_T, and also use these features as prediction targets for RA and TAP below. We also study ImageNet supervised features [7].

Representation Anticipation (RA): As a representative of the self-supervised line of work in predicting a fixed offset into the future, we implement RA [53] on our data and architecture, training a model to predict frozen representations of a network trained for cross-modal correspondence. In vision, we train the network to anticipate one second into the future, while in text, we anticipate the subsequent utterance (on average, $\sim$2 seconds into the future). We train:

$$\underset{\Phi_V, \Phi_T, f_{t+1}}{\arg\min} \; -\cos\Big(f_{t+1}(\Phi_M(M_i)), \Psi_M(M_{i+1})\Big), \quad (10)$$

Time-Agnostic Prediction (TAP): Noting the restrictive nature of the fixed offset formulation, TAP [21] introduces the minimum-across-time formulation to allow the prediction of "bottleneck" predictable moments. We implement their loss, taking the minimum across all future moments in the sampled video segment:

$$\underset{\Phi_V, \Phi_T, f_{t+1}}{\arg\min} \; \min_{i < i' \leq N_M} -\cos\left(f_{t+\Delta}(\Phi_M(M_i)), \Psi_M(M_{i'})\right). \quad (11)$$

While the above two models do not consider the exact same setting as us, we re-implement their approaches as faithfully as possible, training them to predict SOTA features trained for cross-modal correspondence.

MemDPC: In order to efficiently model multiple future hypotheses, MemDPC [16] casts future prediction as estimation of convex combinations of memories stored in a codebook, and achieves SOTA performance on tasks of interest. We evaluate their trained visual future prediction model, which does not take textual information as input.

4.3. Evaluating cycle consistency

Central to our formulation is the model's ability to learn dynamic predictions of the future and past that undo each other, as well as finding strong cross-modal correspondences. Thus, we begin by evaluating how well different model variants are able to solve our self-supervised objective on the Recipes test set. We ablate various design

Choice	Variant	Percentile rank	
		Cycle	Cross-modal
Temporal constraint	None*	-	-
	Similarity loss	93.1	74.4
	Max-index	92.6	74.3
	Max-index + sim. loss	**93.6**	**75.7**
Multi-modal info.	$p_{\text{unimodal}} = 1$	89.8	74.3
	$\mathbf{p_{unimodal} = 0.5}$	**93.6**	**75.7**
	$p_{\text{unimodal}} = 0$	96.5	75.9
Start point selection	**Cross-modal similarity**	**93.6**	**75.7**
	Random	88.7	74.5
Input embedding	**Fine-tuned**	**93.6**	**75.7**
	Frozen cross-modal [35]	67.5	76.8
Cycle path	**Within modalities**	**93.6**	**75.7**
	Across modalities	85.0	73.2
	Chance	50.0	50.0

Table 1. **Cycle-back and cross-model accuracy:** We evaluate models on the percentile rank assigned to ground truth in the cycle and cross-modal tasks on the Recipes test set (100 = ground truth ranked first, 0 = ranked last). Used options are shown **in bold**. *Without any temporal constraint, training collapses.

choices, including multi-modal information usage, cycle edge order, and temporal constraints on edges.

Multi-modal information: As an alternative to defining the state as a learned combination of visual and textual representations $z_\alpha = f(z_{M,\alpha}, z_{M',\alpha})$, we can use only one modality at a time, giving $z_\alpha = z_{M,\alpha}$. The frequency at which only unimodal information is used can be controlled by a hyperparameter p_{unimodal}.

Cycle path: The above formulation navigates between moments in the start modality M, optionally using information from M' to augment representations. We denote this variant **Within modalities**. The order of these edges can also be permuted, such that cycles start in M, retrieve a moment in M', find a future moment in M', then cycle back through M. This variant is denoted **Across modalities**.

Evaluating variants: To compare between different variants, we measure the average percentile rank (*e.g.* 100 = ground truth is ranked first among all candidates, 50 = ranked in the middle, 0 = ranked last) assigned by our model to ground truth cross-modal and cycle nodes. We show this ablation study in Table 1, observing significant gains using our cycle configuration. We hypothesize that across-modality cycles perform worse since switching modalities acts as a bottleneck, forcing the model to discard information that would be useful for subsequent edges.

Visualizing cycles: We show examples of cycles discovered by the trained model in Figure 4. Our model correctly cycles back around 66% of the time (chance is 4%). The model appears to traverse video according to long-term dynamics, as hypothesized. Note that these transitions occur up to one minute apart, highlighting the importance of allowing dynamic prediction offsets.

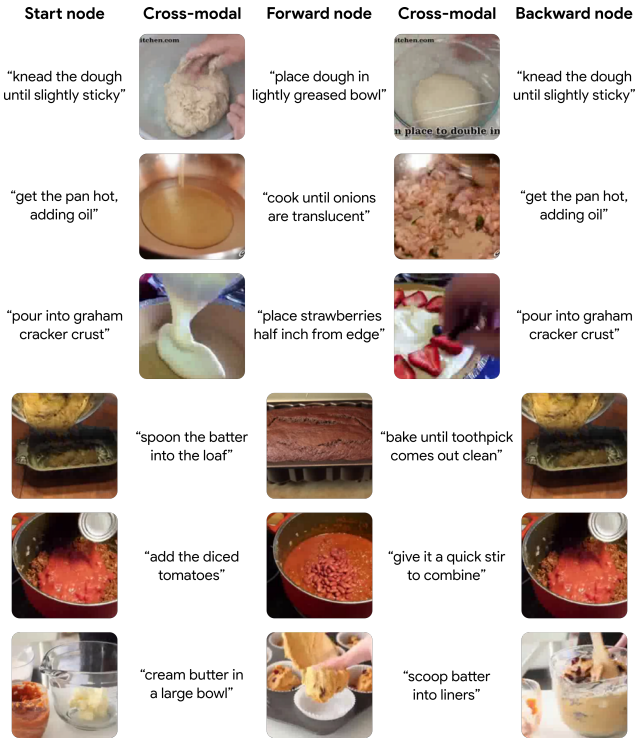

Figure 4. **Emergent long-term temporal dynamics:** We show examples of learned model cycles in the Recipes test set. Given a start node (top: text, bottom: image) sampled as described in Section 3.3, we show the retrieved cross-modal node, the predicted future node and its cross-modal retrieval, and the model's final backward prediction. In the bottom row, we show a failure case, where the forward edge skips too far ahead and breaks the cycle.

4.4. Zero-shot prediction

Ranking transitions by likelihood: To directly evaluate the learned representations π and functions g_{fwd} and g_{back}, we can visualize the pairs of frames (u, v) for which the probability of v being the future of u is highest. We model this probability as the product of the likelihood of states u and v and the forward and backward likelihood of $u \to v$:

$$P_{\text{fwd}}(v|u) = \frac{e^{\pi_{\text{fwd}} \cdot \pi_v}}{\sum_{m \in M} e^{\pi_{\text{fwd}} \cdot \pi_m}}, \quad (12)$$

$$P_{\text{back}}(u|v) = \frac{e^{\pi_{\text{back}} \cdot \pi_u}}{\sum_{m \in M} e^{\pi_{\text{back}} \cdot \pi_m}},$$

$$P(u \to v) = P_{\text{fwd}}(v|u) \cdot P_{\text{back}}(u|v) \cdot P(u) \cdot P(v),$$

where $\pi_{\text{fwd}}, \pi_{\text{back}}$ are the result of running u and v (optionally with cross-modal information) through $g_{\text{fwd}}, g_{\text{back}}$ and $P(x)$ is the concreteness score defined in Equation 7.

We compute this probability efficiently for all n^2 pairs in long clips of continuously sampled video ($n \approx 600$). We then look at the top temporal transitions discovered by the model in each video. We show results on the Recipes test set in Figure 5. The top transitions show clear state transi-

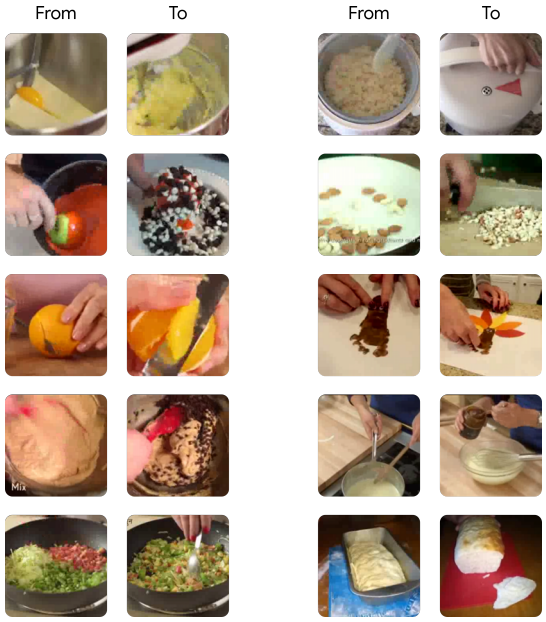

Figure 5. **Discovering transitions in video:** Once trained, the learned model of temporal dynamics can be applied to long (5-10 min.) video sequences to discover the most salient pairwise transitions $u \rightarrow v$. We compute the probability of a transition as defined in Eq. 12 and show the highest-score transitions in 10 different Recipes test set videos.

tions such as adding chocolate chips to dough, segmenting an orange, and baking a loaf. These predictions could not be made by a model trained to predict a fixed future, since they occur at varied temporal offsets.

Predicting future actions: Existing benchmarks *e.g.* [5, 25] focus on predicting action from a few frames in the immediate past or present. Instead, given a few frames, we wish to predict long-term temporal dynamics, which may unfold arbitrarily far into the future. While the former task is more well-defined, the latter is more interesting and relevant. However, ground truth for this task – *i.e.*, per-frame annotation of related future action – is not widely available. We propose using CrossTask task steps as a proxy, since they capture long-term temporal relationships in video.

For a video V belonging to task τ (with N_τ predefined subtasks), let $v \in V$ be a clip with subtask label $T_{\tau,i}$ (i^{th} in the predefined sequence). We would like to predict future actions $T_{\text{future}} = \{T_{\tau,j}\}_{j=i+1}^{N_\tau}$ from v. For example, given a short clip of eggs being added to a bowl with flour, the model should assign high likelihoods to subtasks such as "mix batter" and low likelihoods to "crack eggs" or "season steak". Formally, we define a future likelihood score given a video segment v and candidate future subtask T_j. We first sample frames from the video segment and compute their average embedding $\bar{z}_v$. The likelihood score uses our learned representations and predictive model to define a score $s_{v \rightarrow j} = g_{\text{fwd}}(\bar{z}_v) \cdot \pi_{T_j}$. We compute likelihood scores for all subtask descriptions in the CrossTask validation set, and consider the model's prediction correct if any of the future actions in T_{future} are predicted, since not all future subtasks are necessarily related to the given visual state.

Model	Recall			Percentile rank		
	@ 1	@ 5	@ 10	Worst	Mean	Best
MemDPC* [16]	2.9	15.8	27.4	25.6	48.4	71.4
Cross-modal [35]	2.9	14.2	24.3	28.2	47.9	68.2
Repr. Ant. [53]	3.0	13.3	26.0	25.7	47.7	71.4
TAP [21]	4.5	17.1	27.9	28.3	50.1	71.6
MMCC (ours)	**5.4**	**19.9**	**33.8**	**33.0**	**55.0**	**76.9**

Table 2. **Predicting future actions:** We evaluate models' ability to anticipate action at a high level, potentially minutes into the future, without any fine-tuning. On the CrossTask dataset [68], our model outperforms the previous self-supervised state of the art in inferring possible future actions. *We evaluate MemDPC by clip retrieval since it does not have a textual representation.

Table 2 shows recall and percentile rank statistics for this task. We compare our model to [16, 21, 53], replacing g_{fwd} with each method's predictive model. Since [16] is vision-only, we set π_{T_j} to the average visual representation of all video segments with a given subtask label T_j. We also define a cross-modal similarity score $s_{v \rightarrow j} = \bar{\pi}_v \cdot \pi_{T_j}$ as a strong baseline, taking advantage of contextual similarities in video and text. Our model outperforms all baselines and self-supervised state of the art on detecting the temporal relationships between visual states and future actions.

4.5. Further analysis

Unshuffling bags of frames: The ability to order a shuffled set of states is used to evaluate human language and common-sense reasoning skills, and has been explored as a learning signal in NLP [14, 30, 31]. This same ability can also be used to discover temporal structure and summaries of events from large image datasets, as in [23]. We solve this problem by finding the optimal explanation of shuffled video given by iterative application of our temporal dynamics model. Out of all $n!$ possible orderings $\{i_1, i_2, \ldots, i_n\}$, we select the one for which $\prod_j P(x_{i_j} \rightarrow x_{i_j+1})$ is highest.

Given $P(u \rightarrow v)$ scores computed by Eq. 12, we induce a fully-connected directed graph with sampled frames as nodes and edge weights given by $\texttt{weight}(\vec{uv}) \equiv -\log P(u \rightarrow v)$. Adding a special null node connected to all other nodes with edge weight 0 allows running this graph through an off-the-shelf traveling salesperson problem (TSP) solver[3]. The optimal TSP solution then represents the lowest-cost (ordered) path through all video clips, effectively unshuffling the input.

We run this experiment on CrossTask, where videos are annotated with ordered steps and their associated temporal

[3] https://pypi.org/project/elkai/

Figure 6. **Unshuffling image collections:** We show example video sequences in the ordering given by treating the induced graph of log-likelihoods as an instance of the traveling salesperson problem. We list the ground truth index in the sequence under each clip. Even accepting only sparse video frames as input, our model makes reasonable predictions on this challenging task.

Model	Kendall's τ (↑)	Spearman's ρ (↑)	Edit dist. (↓)
Chance	0.0000	0.0000	6.5822
Repr. Ant. [54]	0.3383	0.4132	5.4596
MemDPC [16]	0.3492	0.4206	5.3398
TAP [21]	0.3344	0.4107	5.4178
MMCC (ours)	**0.3632**	**0.4420**	**5.3343**
MMCC (vision only)	0.3530	0.4328	5.3370

Table 3. **Unshuffling image collections:** As defined in Eq. 12, $\log P(u \rightarrow v)$ gives a log-likelihood score of v being the future state of u. These scores induce a graph which is optimally traversed using a TSP solver. Each model above defines a different P and is applied to shuffled videos from the Recipes test set. Our model outperforms previous state of the art on all metrics.

segments. We treat each segment as a node by computing its average visual representation, as before. We then use these representations to find $P(u \rightarrow v)$ scores between labeled segments and solve an optimal path. We run this experiment both in vision only (by passing projected visual representations directly into g in Eq. 12) as well as with ground truth vision-text pairings, and show results in Table 3. We show example predicted orderings in Figure 6. Again, we can replace g_{fwd} with the future prediction model in other methods and run the same algorithm. Our model outperforms previous work on all evaluation metrics.

Discovering the arrow of time: To further examine whether our model has learned to discover meaningful transitions between states, we explore the arrow of time classification task, introduced in [38, 40, 59]. In [59], a network is trained on short videos (on the order of seconds) to predict whether input is being played forward or backward.

		Sampling strategy					
		Rand	**Cos sim**	**TAP**	**RA**	**Model**	***Avg***
Features	Random	50.4	50.7	51.3	51.4	51.2	51.0
	ImageNet	51.5	52.1	50.8	50.9	53.4	51.7
	Cross-modal	52.6	53.3	50.9	50.6	55.8	52.6
	Repr. Ant. [53]	50.7	51.4	51.2	51.2	51.7	51.2
	TAP [21]	50.8	51.4	51.4	51.3	51.8	51.3
	MMCC (ours)	52.3	53.4	50.5	50.7	**69.2**	**55.2**
	Average	51.4	52.0	51.0	51.0	**55.5**	52.2
	Chance	50.0	50.0	50.0	50.0	50.0	50.0
	From scratch	51.1	52.1	51.8	51.4	62.5	53.8

Table 4. **Learning the arrow of time:** We train a linear layer on frozen features as well as a full model from scratch (last row) to detect which of two input frames comes first. This task is near-impossible when sampling random frames. Using our temporal model to sample frames leads to a significant improvement in performance, indicating that our model can identify salient transitions in video. Our visual embedding also outperforms other models, highlighting the importance of temporally-attuned representations.

We consider the more challenging task of predicting the temporal relationship between two far-apart input frames – which one comes first? For frames which depict unrelated moments, this task is perhaps near-impossible, even for humans. But if frames show semantically related states, the direction of likely transition provides a useful signal for solving the arrow-of-time task.

We train linear classifiers on top of frozen features as well as a full network from scratch to solve the arrow of time task on randomly shuffled pairs of frames. We sample pairs of frames using our learned predictive model by selecting the highest-probability futures of start frames selected with the concreteness prior (Eq. 7). We demonstrate in Table 4 that the temporal ordering of frames mined by our model is much more classifiable than that of frames sampled using predictive models in previous work. Further, our learned features are much more able to classify a given pair of frames, since they must capture temporal information in training. This confirms that a strong understanding of dynamics that emerges from the cycle consistency task.

5. Conclusion

We introduce a self-supervised method to learn temporal dynamics by cycling through narrated video. Despite the simplicity of our architecture, our model is able to discover long-term state transitions in vision and language. We show that this model can be applied without further training to challenging downstream tasks such as anticipating far-away action and ordering collections of image data.

References

[1] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *CVPR*, 2016. 2

[2] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 2

[3] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv:1710.11252*, 2017. 2

[4] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. Recycle-GAN: Unsupervised video retargeting. In *ECCV*, 2018. 2

[5] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The EPIC-Kitchens dataset. In *ECCV*, 2018. 2, 5, 7

[6] Virginia R de Sa. Learning classification with unlabeled data. In *NeurIPS*, 1994. 2

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[8] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. *arXiv:1802.07687*, 2018. 2

[9] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *CVPR*, 2019. 3

[10] Frederik Ebert, Chelsea Finn, Alex X Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. *arXiv:1710.05268*, 2017. 1

[11] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *arXiv:1605.07157*, 2016. 2

[12] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *ICRA*, 2017. 1

[13] Antonino Furnari and Giovanni Maria Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In *CVPR*, 2019. 2

[14] Jingjing Gong, Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. End-to-end neural sentence ordering using pointer network. *arXiv:1611.04953*, 2016. 7

[15] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *ICCV Workshops*, 2019. 1, 2

[16] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. *arXiv:2008.01065*, 2020. 1, 2, 5, 7, 8

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 5

[18] Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. In *NeurIPS*, 2018. 2

[19] De-An Huang and Kris M Kitani. Action-reaction: Forecasting the dynamics of human interaction. In *ECCV*, 2014. 2

[20] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020. 3

[21] Dinesh Jayaraman, Frederik Ebert, Alexei A Efros, and Sergey Levine. Time-agnostic prediction: Predicting predictable video frames. In *ICLR*, 2019. 2, 5, 7, 8

[22] Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In *ICCV*, 2015. 1

[23] Gunhee Kim, Leonid Sigal, and Eric P Xing. Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In *CVPR*, 2014. 7

[24] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *ECCV*, 2012. 2

[25] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*, 2014. 5, 7

[26] Nilesh Kulkarni, Abhinav Gupta, and Shubham Tulsiani. Canonical surface mapping via geometric cycle consistency. In *ICCV*, 2019. 3

[27] Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. Videoflow: A conditional flow-based model for stochastic video generation. *arXiv:1903.01434*, 2019. 2

[28] Tian Lan, Tsung-Chuan Chen, and Silvio Savarese. A hierarchical representation for future action prediction. In *ECCV*, 2014. 2

[29] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv:1804.01523*, 2018. 1, 2

[30] Haejun Lee, Drew A Hudson, Kangwook Lee, and Christopher D Manning. SLM: Learning a discourse language representation with sentence unshuffling. *arXiv:2010.16249*, 2020. 7

[31] Lajanugen Logeswaran, Honglak Lee, and Dragomir Radev. Sentence ordering and coherence modeling using recurrent neural networks. *arXiv:1611.02654*, 2016. 7

[32] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv:1605.08104*, 2016. 2

[33] Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy. What's cookin'? interpreting cooking videos using text, speech and vision. *arXiv:1503.01558*, 2015. 2

[34] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv:1511.05440*, 2015. 1

[35] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020. 2, 3, 4, 5, 6, 7

[36] Antoine Miech, Ivan Laptev, Josef Sivic, Heng Wang, Lorenzo Torresani, and Du Tran. Leveraging the present to anticipate the future in videos. In *CVPR Workshops*, 2019. 2

[37] 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*, 2019. 2, 5

[38] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016. 8

[39] Nemanja Petrovic, Aleksandar Ivanovic, and Nebojsa Jojic. Recursive estimation of generative models of video. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 79–86. IEEE, 2006. 1

[40] Lyndsey C Pickup, Zheng Pan, Donglai Wei, YiChang Shih, Changshui Zhang, Andrew Zisserman, Bernhard Scholkopf, and William T Freeman. Seeing the arrow of time. In *CVPR*, 2014. 8

[41] Silvia L Pintea, Jan C van Gemert, and Arnold WM Smeulders. Déja vu. In *ECCV*, 2014. 2

[42] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv:1412.6604*, 2014. 2

[43] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *CVPR*, 2015. 2

[44] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Chris Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *IJCV*, 2017. 2

[45] Mohammad Sadegh Aliakbarian, Fatemeh Sadat Saleh, Mathieu Salzmann, Basura Fernando, Lars Petersson, and Lars Andersson. Encouraging LSTMs to anticipate actions very early. In *ICCV*, 2017. 2

[46] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using LSTMs. In *ICML*, 2015. 2

[47] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv:1906.05743*, 2019. 1, 2

[48] Chen Sun, Per Karlsson, Jiajun Wu, Joshua B Tenenbaum, and Kevin Murphy. Stochastic prediction of multi-agent interactions from partial observations. *arXiv:1902.09641*, 2019. 2

[49] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019. 2

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3

[51] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *arXiv:1706.08033*, 2017. 2

[52] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to generate long-term future via hierarchical prediction. *arXiv:1704.05831*, 2017. 2

[53] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *CVPR*, 2016. 1, 2, 5, 7, 8

[54] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *NeurIPS*, 2016. 1, 2, 8

[55] Carl Vondrick and Antonio Torralba. Generating the future with adversarial transformers. In *CVPR*, 2017. 2

[56] Jacob Walker, Abhinav Gupta, and Martial Hebert. Patch to the future: Unsupervised visual prediction. In *CVPR*, 2014. 2

[57] Jacob Walker, Abhinav Gupta, and Martial Hebert. Dense optical flow prediction from a static image. In *ICCV*, 2015. 2

[58] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019. 3

[59] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *CVPR*, 2018. 8

[60] Yu Wu, Linchao Zhu, Xiaohan Wang, Yi Yang, and Fei Wu. Learning to anticipate egocentric actions by imagination. *IEEE Transactions on Image Processing*, 30:1143–1152, 2020. 2

[61] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, 2016. 2

[62] Tianfan Xue, Jiajun Wu, Katherine Bouman, and Bill Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NeurIPS*, 2016. 2

[63] Shoou-I Yu, Lu Jiang, and Alexander Hauptmann. Instructional videos for unsupervised harvesting and learning of action examples. In *ACM MM*, 2014. 2

[64] Jenny Yuen and Antonio Torralba. A data-driven approach for event prediction. In *ECCV*, 2010. 2

[65] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 2

[66] Tinghui Zhou, Philipp Krähenbühl, Mathieu Aubry, Qixing Huang, and Alexei A. Efros. Learning dense correspondence via 3D-guided cycle consistency. In *CVPR*, 2016. 2

[67] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2

[68] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *CVPR*, 2019. 5, 7